AIX 5L Version 5.2

Commands Reference, Volume 4

AIX 5L Version 5.2

Commands Reference, Volume 4

Note
Before using this information and the product it supports, read the information in “Notices” on page 667.

Fourth Edition (October 2002)

This edition applies to AIX 5L Version 5.2 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 H6DS-905-6C006, 11501 Burnet Road, Austin, Texas 78758-3493. To send comments electronically, use this commercial Internet address: aix6kpub@austin.ibm.com. Any information that you supply may be used without incurring any obligation to you.

© Copyright International Business Machines Corporation 1997, 2002. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About This Book xi
Who Should Use This Book xi
How to Use This Book xi
Highlighting xi
ISO 9000 xiv
32-Bit and 64-Bit Support for the UNIX98 Specification xiv
Related Information xiv

Alphabetical Listing of Commands 1
named Daemon 1
named4 Daemon 2
named8 Daemon 4
named9 Daemon 8
namerslv Command 9
ncheck Command 12
ndp Command 13
ndpd-host Daemon 13
ndpd-router Daemon 14
ndx Command 19
neqn Command 20
netpmon Command 21
netstat Command 30
newaliases Command 40
newform Command 40
newgrp Command 43
newkey Command 44
news Command 45
next Command 47
nfsd Daemon 48
nfso Command 49
nfsstat Command 64
nice Command 67
nim Command 69
nimadm Command 79
nimclient Command 84
nimconfig Command 87
nimdef Command 89
niminit Command 93
nis_cachemgr Daemon 95
nisaddcred Command 97
nisaddent Command 100
niscat Command 103
nischgrp Command 104
nischmod Command 106
nischown Command 107
nischttl Command 109
nisclient Command 110
nisdefaults Command 113
niserror Command 115
nisgrep Command 116
nisgrpadm Command 117
nisinit Command 118
nisln Command 120

© Copyright IBM Corp. 1997, 2002

nislog Command . . . 121
nisls Command . . . 122
nismatch Command . . . 123
nismkdir Command . . . 124
nismkuser Command . . . 126
nisping Command . . . 128
nispopulate Command . . . 130
nisrm Command . . . 132
nisrmdir Command . . . 133
nisrmuser Command . . . 134
nisserver Command . . . 135
nissetup Command . . . 137
nisshowcache Command . . . 138
nisstat Command . . . 138
nistbladm Command . . . 139
nistest Command . . . 143
nistoldif Command . . . 144
nisupdkeys Command . . . 146
nl Command . . . 147
nlssrc Command . . . 149
nm Command . . . 151
no Command . . . 154
nohup Command . . . 191
notifyevent Command . . . 193
nrglbd Daemon . . . 195
nroff Command . . . 195
nslookup Command . . . 198
nsupdate Command . . . 204
nsupdate8 Command . . . 206
nsupdate9 Command . . . 208
ntpdate Command . . . 211
ntpq Command . . . 212
ntptrace Command . . . 217
ntsc Command . . . 218
nulladm Command . . . 219
number Command . . . 220
od Command . . . 221
odmadd Command . . . 225
odmchange Command . . . 227
odmcreate Command . . . 227
odmdelete Command . . . 230
odmdrop Command . . . 231
odmget Command . . . 231
odmshow Command . . . 232
on Command . . . 233
oslevel Command . . . 234
ospf_monitor Command . . . 235
pac Command . . . 237
pack Command . . . 238
packf Command . . . 240
pagdel Command . . . 242
pagesize Command . . . 242
paginit Command . . . 243
paglist Command . . . 244
panel20 Command . . . 244
passwd Command . . . 245

paste Command 248
patch Command 250
pathchk Command 255
pax Command 256
pcat Command 268
pdelay Command 269
pdisable Command 269
penable Command 270
pg Command 271
phold Command 274
pic Command 275
pick Command 281
ping Command 284
pioattred Command 289
piobe Command 290
pioburst Command 292
piocnvt Command 294
piodigest Command 295
piodmgr Command 296
piofontin Command 297
pioformat Command 298
piofquote Command 301
piolsvp Command 301
piomgpdev Command 304
piomkapqd Command 305
piomkpq Command 307
piomsg Command 309
pioout Command 310
piopredef Command 312
pkgadd Command 314
pkgask Command 316
pkgchk Command 318
pkginfo Command 319
pkgmk Command 321
pkgparam Command 323
pkgproto Command 325
pkgrm Command 327
pkgtrans Command 328
plotgbe Command 329
plotlbe Command 331
pmcycles Command 332
pop3d Daemon 333
portmap Daemon 334
portmir Command 335
post Command 337
pppattachd Daemon 338
pppcontrold Daemon 340
pppdial Command 344
pppstat Command 345
pprof Command 347
pr Command 348
prctmp Command 351
prdaily Command 352
preprpnode Command 352
prev Command 355
printenv Command 356

printf Command 357
proccred Command 360
processPtf Command 361
procfiles Command 363
procflags Command 364
procldd Command 366
procmap Command 367
procrun Command 368
procsig Command 369
procstack Command 370
procstop Command 371
proctree Command 372
procwait Command 374
procwdx Command 375
prof Command 376
proff Command 378
prompter Command 379
proto Command 380
proxymngr Command 381
prs Command (SCCS) 383
prtacct Command 386
prtconf Command 387
ps Command 391
ps4014 Command 407
ps630 Command 408
psc or psdit Command 410
pshare Command 412
psplot Command 413
psrev Command 414
psroff Command 414
pstart Command 417
pstat Command 418
ptfins command 419
ptfpkg Command 420
ptx Command 424
pwchange Command 426
pwck Command 428
pwd Command 428
pwdadm Command 429
pwdck Command 431
pwtokey Command 434
pxed Command 437
qadm Command 438
qcan Command 439
qchk Command 441
qdaemon Command 442
qhld Command 443
qmov Command 445
qosadd Command 446
qoslist Command 447
qosmod Command 448
qosremove Command 449
qosstat Command 450
qpri Command 451
qprt Command 452
qstatus Command 460

quiz Command 462
quot Command 463
quota Command 465
quotacheck Command 466
quotaon or quotaoff Command 467
ranlib Command 469
ras_logger Command 469
rc Command 471
rc.mobip6 Command 471
rc.powerfail Command 473
rcp Command 474
rcvdist Command 478
rcvpack Command 479
rcvstore Command 479
rcvtty Command 481
rdist Command 481
rdistd Command 494
rdump Command 495
read Command 497
readlvcopy Command 499
reboot or fastboot Command 499
recreatevg Command 501
redefinevg Command 503
reducevg Command 503
refer Command 505
refile Command 507
refresh Command 509
refrsrc Command 510
regcmp Command. 512
rembak Command. 513
remove Command. 515
renice Command 515
reorgvg Command 517
repl Command 518
replacepv Command 522
repquota Command 523
reset Command 524
resize Command 525
restart-secldapclntd Command 526
restbase Command 527
restore Command 528
restorevgfiles Command 537
restvg Command 538
rev Command 540
revnetgroup Command 541
rexd Daemon 542
rexec Command 542
rexecd Daemon 544
rgb Command 545
ripquery Command 546
rlogin Command 547
rlogind Daemon 549
rm Command 551
rm_niscachemgr Command 553
rm_nisd Daemon 554
rm_nispasswdd Daemon 556

rmail Command . . . 557
rmaudrec Command . . . 557
rmC2admin Command . . . 561
rmCCadmin Command . . . 562
rmcctrl Command . . . 563
rmclass Command . . . 565
rmcomg Command . . . 566
rmcondition Command . . . 568
rmcondresp Command . . . 570
rmdel Command . . . 573
rmdev Command . . . 574
rmdir Command . . . 575
rmf Command . . . 577
rmfilt Command . . . 578
rmfs Command . . . 578
rmgroup Command . . . 580
rmitab Command . . . 581
rmkeyserv Command . . . 581
rmlv Command . . . 583
rmlvcopy Command . . . 584
rmm Command . . . 585
rmnamsv Command . . . 586
rmnfs Command . . . 587
rmnfsexp Command . . . 588
rmnfsmnt Command . . . 589
rmnotify Command . . . 590
rmpath Command . . . 591
rmprtsv Command . . . 593
rmps Command . . . 594
rmqos Command . . . 595
rmque Command . . . 595
rmquedev Command . . . 596
rmramdisk Command . . . 597
rmresponse Command . . . 598
rmrole Command . . . 600
rmrpdomain Command . . . 601
rmrpnode Command . . . 604
rmrset Command . . . 605
rmrsrc Command . . . 606
rmsensor Command . . . 608
rmserver Command . . . 610
rmsock Command . . . 611
rmss Command . . . 611
rmssys Command . . . 614
rmt Command . . . 615
rmtcpip Command . . . 616
rmtun Command . . . 617
rmuser Command . . . 617
rmvfs Command . . . 619
rmvirprt Command . . . 620
rmyp Command . . . 621
rndc Command . . . 622
rndc-confgen Command . . . 623
roffbib Command . . . 624
route Command . . . 624
routed Daemon . . . 628

rpc.nisd Daemon 632
rpc.nispasswdd Daemon 634
rpc.pcnfsd Daemon 635
rpcgen Command 637
rpcinfo Command 638
rrestore Command 640
Rsh Command 643
rsh or remsh Command. 645
rshd Daemon 648
rstatd Daemon 650
rtl_enable Command. 651
runacct Command. 653
runcat Command 656
rup Command 656
ruptime Command. 657
ruser Command 659
rusers Command 660
rusersd Daemon 661
rwall Command. 662
rwalld Daemon 663
rwho Command 664
rwhod Daemon 665

Appendix. Notices 667
Trademarks 668

Index 669

About This Book

This book is Volume 4 of the six-volume *AIX 5L Version 5.2 Commands Reference*, which contains reference information on operating system commands. It describes the tasks each command performs, how commands can be modified, how they handle input and output, who can run them, and provides a master index for all six volumes.

For a quick reference list of commands arranged in functional groups, see Volume 6.

Who Should Use This Book

This book is intended for users of operating system commands.

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.

© Copyright IBM Corp. 1997, 2002

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.

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 UNIX98 Specification

Beginning with Version 4.3, the operating system is designed to support The Open Group's UNIX98 Specification 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 4.3 even more open and portable for applications.

At the same time, compatibility with previous releases of the operating system is preserved. This is accomplished by the creation of a new environment variable, which can be used to set the system environment on a per-system, per-user, or per-process basis.

To determine the proper way to develop a UNIX98-portable application, you may need to refer to The Open Group's UNIX98 Specification, which can be obtained on a CD-ROM by ordering *Go Solo 2: The Authorized Guide to Version 2 of the Single UNIX Specification*, a book which includes The Open Group's UNIX98 Specification on a CD-ROM.

Related Information

The following books contain information about or related to commands:

- *AIX 5L Version 5.2 Commands Reference, Volume 1*
- *AIX 5L Version 5.2 Commands Reference, Volume 2*
- *AIX 5L Version 5.2 Commands Reference, Volume 3*
- *AIX 5L Version 5.2 Commands Reference, Volume 4*
- *AIX 5L Version 5.2 Commands Reference, Volume 5*
- *AIX 5L Version 5.2 Commands Reference, Volume 6*
- *AIX 5L Version 5.2 Files Reference*
- *AIX 5L Version 5.2 Glossary*
- *AIX 5L Version 5.2 Guide to Printers and Printing*
- *AIX 5L Version 5.2 Installation Guide and Reference*
- *AIX 5L Version 5.2 AIX Installation in a Partitioned Environment*
- *AIX 5L Version 5.2 Network Information Services (NIS and NIS+) Guide*
- *AIX 5L Version 5.2 Operating System Installation: Getting Started*
- *AIX 5L Version 5.2 Performance Management Guide*
- *AIX 5L Version 5.2 Performance Tools Guide and Reference*
- *AIX 5L Version 5.2 Security Guide*
- *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices*
- *AIX 5L Version 5.2 System Management Guide: Communications and Networks*
- *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*
- *AIX 5L Version 5.2 System User's Guide: Communications and Networks*
- *AIX 5L Version 5.2 System User's Guide: Operating System and Devices*
- *AIX 5L Version 5.2 Technical Reference: Base Operating System and Extensions Volume 1*

- *AIX 5L Version 5.2 Technical Reference: Base Operating System and Extensions Volume 2*
- *AIX 5L Version 5.2 Technical Reference: Communications Volume 1*
- *AIX 5L Version 5.2 Technical Reference: Communications Volume 2*
- *AIX 5L Version 5.2 Technical Reference: Kernel and Subsystems Volume 1*
- *AIX 5L Version 5.2 Technical Reference: Kernel and Subsystems Volume 2*
- *AIX 5L Version 5.2 Web-based System Manager Administration Guide*
- *Performance Toolbox Version 2 and 3 for AIX: Guide and Reference*

Alphabetical Listing of Commands

named Daemon

Purpose

Provides the server function for the Domain Name Protocol.

Syntax

Refer to the syntax for either the **named4**, **named8**, or the **named9** daemon.

Description

AIX supports three versions of BIND: 4, 8, and 9. By default, **named** links to **named4**, **nsupdate** to **nsupdate4**, **named-xfer** to **named-xfer4**. To use a different version of **named**, you must relink the symbolic links accordingly for the **named** and **named-xfer** daemons.

For example, to use **named8**:

```
ln -fs /usr/sbin/named8 /usr/sbin/named
ln -fs /usr/sbin/named8-xfer /usr/sbin/named
```

nsupdate4 can be use with **named8**, but **nsupdate9** must be used with **named9** because the security process is different. It does not matter what **named-xfer** is linked to when using **named9** because the daemon does not use it.

Files

/usr/sbin/named	Contains the **named** daemon.
/usr/sbin/named4	Contains the **named4** daemon.
/usr/sbin/named8	Contains the **named8** daemon.
/usr/sbin/named9	Contains the **named9** daemon.
/etc/resolv.conf	Specifies the use of domain name services.
/etc/rc.tcpip	Initializes daemons at each system restart.
/etc/named.pid	Stores process IDs.
/etc/services	Defines socket service assignments.
/usr/samples/tcpip/named.boot	Contains the sample **named.boot** file with directions for its use.
/usr/samples/tcpip/named.data	Contains the sample DOMAIN data file with directions for its use.
/usr/samples/tcpip/hosts.awk	Contains the sample **awk** script for converting an **/etc/hosts** file to an **/etc/named.data** file. This file also contains directions for its use.
/usr/samples/tcpip/addrs.awk	Contains the sample **awk** script for converting an **/etc/hosts** file to an **/etc/named.rev** file. This file also contains directions for its use.
/usr/samples/tcpip/named.dynamic	Contains a dynamic database setup.

Related Information

The **nslookup** command, **traceroute** command, **kill** command.

The **named4**, **named8** and **named9** daemons.

The **named.conf** file format, **DOMAIN Cache** file format, **DOMAIN Data** file format, **DOMAIN Reverse Data** file format, **DOMAIN Local Data** file format, **resolv.conf** file format.

© Copyright IBM Corp. 1997, 2002

named4 Daemon

Purpose

Provides the server function for the Domain Name Protocol.

Syntax

/usr/sbin/named4 [**-d** *DebugLevel*] [**-p** *PortNumber*] [**-b** *BootFile*]

Description

The **/usr/sbin/named4** daemon is the server for the Domain Name Protocol (DOMAIN). The **named4** daemon runs on name-server hosts and controls the domain-name resolution function.

This operating system provides two name server daemons, the **named4** daemon and the **named8** daemon. While both provide the same service of domain-name resolution, the **named8** daemon supports the more robust configuration file **named.conf**. The **named4** daemon is provided and enabled by default for backwards compatibility as it is configured with the more simplistic **named.boot** file.

Selection of which name server daemon to use is controlled by the **/usr/sbin/named** and **/usr/sbin/named-xfer** symbolic links. By default, these are links to the **named4** and **named4-xfer** executables, but can be changed to reference the **named8** and **named8-xfer** executables. Note that it is required that both symbolic links are changed in tandem, as the version 4 name server is not compatible with the version 8 zone transfer, and visa-versa.

> **Note:**The **named4** daemon can be controlled using the System Resource Controller (SRC) or the System Management Interface Tool (SMIT). Use the **rc.tcpip** file to start the daemon with each system startup.

The**named4**daemon listens for name-server requests generated by resolver routines running on foreign hosts. The daemon listens to the socket defined in the **/etc/services** file; the entry in the **/etc/services** file begins with `domain` . However, this socket assignment can be overridden using the **-p**flag on the command line.

> **Note:**The **/etc/resolv.conf** file tells the local kernel and resolver routines to use the DOMAIN protocol. The **/etc/resolv.conf** file must exist and contain either the local host's address or the loopback address (127.0.0.1), or be empty to use the **named4** daemon on the DOMAIN name server host. If the **/etc/resolv.conf** file does not exist, the local kernel and resolver routines use the **/etc/hosts** database. When this occurs, the **named4**daemon does not function properly.

Manipulating the named4 Daemon with the System Resource Controller

The **named4** daemon is a subsystem controlled by the System Resource Controller (SRC). The **named4** daemon is a member of the **tcpip** system group. This daemon is disabled by default and can be manipulated by the following SRC commands:

startsrc	Starts a subsystem, group of subsystems, or a subserver.
stopsrc	Stops a subsystem, group of subsystems, or a subserver.
refresh	Causes the **named4**daemon to reread the **/etc./named.boot** file. Depending on the contents of the file, the **refresh** command may or may not reload the database.
traceson	Enables tracing of a subsystem, group of subsystems, or a subserver.
tracesoff	Disables tracing of a subsystem, group of subsystems, or a subserver.
lssrc	Gets the status of a subsystem, group of subsystems, or a subserver.

Flags

-b*BootFile*	Specifies an alternate boot file.
-d*DebugLevel*	Provides a debugging option. The **-d** flag causes the **named4**daemon to write debugging information to a file named **/var/tmp/named.run**. The *DebugLevel* variable determines the level of messages printed, with valid levels from 1 to 11, where level 11 supplies the most information.
-p*PortNumber*	Reassigns the Internet socket where the**named4**daemon listens for DOMAIN requests. If this variable is not specified, the**named4**daemon listens to the socket defined in the **/etc/services** file; the entry in the **/etc/services** file begins with `domain .`

Signals

The following signals have the specified effect when sent to the **named4** daemon process using the **kill** command:

SIGHUP	The**named4** daemon rereads the **/etc/named.boot**file. Depending on the contents of the file, the **SIGHUP** signal may or may not reload the database.
SIGINT	The **named4** daemon dumps the current database to a file named**/var/tmp/named_dump.db**. In the dump file, names with the label **name error** indicate negative cache entries. This happens when a server responds that the specified domain name does not exist. Names labeled as **data error** also indicate negative cache entries. This happens when a server responds that there are no records of the specified type for the (valid) domain name.
SIGUSR1	The**named4** daemon turns on debugging; each subsequent **SIGUSR1** signal increments the debugging level. The debugging information is written to the **/var/tmp/named.run** file.
SIGUSR2	The**named4** daemon turns off debugging.
SIGABRT/SIGIOT	The**named4** daemon dumps the current statistics to the **/var/tmp/named.stat** file.

Examples

1. To start the **named4** daemon normally, enter the following:

   ```
   startsrc -s named
   ```

   This command starts the daemon. You can use this command in the **rc.tcpip** file or on the command line. The **-s** flag specifies that the subsystem that follows is to be started. The process ID of the **named4** daemon is stored in the **/etc/named.pid** file upon startup.
2. To stop the **named4** daemon normally, enter:

   ```
   stopsrc -s named
   ```

   This command stops the daemon. The **-s** flag specifies that the subsystem that follows is to be stopped.
3. To get short status from the **named4** daemon, enter:

   ```
   lssrc -s named
   ```

   This command returns the name of the daemon, the process ID of the daemon, and the state of the daemon (active or inactive).
4. To enable debugging for the **named4** daemon, enter:

   ```
   traceson -s named
   ```

   OR

```
kill -30 `cat /etc/named.pid`
```

The**named4** daemon turns on debugging in response to either of these commands; each subsequent command increments the debugging level. The debugging information is written to the **/var/tmp/named.run** file.

5. To turn off debugging for the **named4** daemon, enter:

   ```
   tracesoff
   ```

   OR

   ```
   kill -35 `cat /etc/named.pid`
   ```

   Either of these commands immediately turns off all debugging.
6. To start the **named4** daemon at the highest debugging level using the **startsrc** command, enter the following:

   ```
   startsrc -s named -a -d 11
   ```

   This command writes debugging messages to the **/var/tmp/named.run** file.

Files

/usr/sbin/named4	Contains the **named** daemon.
/etc/resolv.conf	Specifies the use of domain name services.
/etc/rc.tcpip	Initializes daemons at each system restart.
/etc/named.pid	Stores process IDs.
/etc/services	Defines socket service assignments.
/usr/samples/tcpip/named.boot	Contains the sample **named.boot** file with directions for its use.
/usr/samples/tcpip/named.data	Contains the sample DOMAIN data file with directions for its use.
/usr/samples/tcpip/hosts.awk	Contains the sample **awk** script for converting an **/etc/hosts** file to an **/etc/named.data** file. This file also contains directions for its use.
/usr/samples/tcpip/addrs.awk	Contains the sample **awk** script for converting an **/etc/hosts** file to an **/etc/named.rev** file. This file also contains directions for its use.
/usr/samples/tcpip/named.dynamic	Contains a dynamic database setup.

Related Information

The **nslookup** command, **traceroute** command, **kill** command.

The **named8** daemon.

The **named.conf** file format, **DOMAIN Cache** file format, **DOMAIN Data** file format, **DOMAIN Reverse Data** file format, **DOMAIN Local Data** file format, **resolv.conf** file format.

TCP/IP Name Resolution and TCP/IP Daemons in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

Configuring Name Servers and Planning for DOMAIN Name Resolution in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

---

named8 Daemon

Purpose

Provides the server function for the Domain Name Protocol.

Syntax

/usr/sbin/named8 [**-d** *DebugLevel*] [**-p** *PortNumber*] [**-c** *ConfFile*] [**-w** *WorkingDirectory*] [**-t** *RootDirectory*] [**-q**] [**-r**] [**-f**]

Description

The **/usr/sbin/named8** daemon is the server for the Domain Name Protocol (DOMAIN). The **named8** daemon runs on name server hosts and controls the domain-name resolution function.

This operating system provides two name server daemons, the **named4** daemon and the **named8** daemon. While both provide the same service of domain-name resolution, the **named8** daemon supports the more robust configuration file **named.conf**. The **named8** daemon is also more current, supporting newer specifications, and containing many security fixes. The **named4** daemon is being phased out of service in support of the **named8** daemon, and as a result, users are encouraged to migrate their name server installations to the newer configuration format.

Selection of which name server daemon to use is controlled by the **/usr/sbin/named** and **/usr/sbin/named-xfer** symbolic links. By default, these are links to the **named4** and **named4-xfer** executables, but can be changed to reference the **named8** and **named8-xfer** executables. Note that it is required that both symbolic links are changed in tandem, as the version 4 name server is not compatible with the version 8 zone transfer, and vice-versa.

> **Note:** The **named8** daemon can be controlled using the System Resource Controller (SRC) or the System Management Interface Tool (SMIT). Use the **rc.tcpip** file to start the daemon with each system startup.

The **named8** daemon listens for name-server requests generated by resolver routines running on foreign hosts. The daemon listens to the socket defined in the **/etc/services** file; the entry in the **/etc/services** file begins with `domain`. However, this socket assignment can be overridden using the **-p** flag on the command line.

> **Note:** The **/etc/resolv.conf** file tells the local kernel and resolver routines to use the DOMAIN protocol. The **/etc/resolv.conf** file must exist and contain either the local host's address or the loopback address (127.0.0.1) to use the **named8** daemon on the DOMAIN name server host. If the **/etc/resolv.conf** file does not exist, the local kernel and resolver routines use the **/etc/hosts** database. When this occurs, the **named8** daemon does not function properly.

Manipulating the named8 Daemon with the System Resource Controller

The **named8** daemon is a subsystem controlled by the System Resource Controller (SRC). The **named8** daemon is a member of the **tcpip** system group. This daemon is disabled by default and can be manipulated by the following SRC commands:

startsrc	Starts a subsystem, group of subsystems,or a subserver.
stopsrc	Stops a subsystem, group of subsystems, or a subserver.
refresh	Causes the **named8** daemon to reread the **/etc/named.conf** file. Depending on the contents of the file, the **refresh** command may or may not reload the listed databases.
traceson	Enables tracing of a subsystem, group of subsystems, or a subserver.
tracesoff	Disables tracing of a subsystem, group of subsystems, or a subserver.
lssrc	Gets the status of a subsystem, group of subsystems, or a subserver.

Flags

-b | **-c***ConfFile*	Specifies an alternate configuration file.

-d*DebugLevel*	Provides a debugging option. The **-d** flag causes the **named8** daemon to write debugging information to a file named by default **/var/tmp/named.run**. The *DebugLevel* variable determines the level of messages printed, with valid levels from 1 to 11, where level 11 supplies the most information.
-p*PortNumber*	Reassigns the Internet socket where the **named8** daemon listens for DOMAIN requests. If this variable is not specified, the **named8** daemon listens to the socket defined in the **/etc/services** file; the entry in the **/etc/services** file begins with `domain`.
-w*WorkingDirectory*	Changes the working directory of the **named8** daemon. This option can be specified or overridden by the ″directory″ configuration option.
-t*RootDirectory*	Specifies a directory to be the new root directory for the **named8** daemon using the **chroot** command.
-q	Enables logging of all name service queries.
-r	Disables the server's ability to recurse and resolve queries outside of the server's local databases.
-f	Indicates to run the name server daemon in the foreground rather than becoming a background job.

Signals

The following signals have the specified effect when sent to the **named8** daemon process using the **kill** command:

SIGHUP	The **named8** daemon rereads the **/etc/named.conf**file. Depending on the contents of the file, the **SIGHUP** signal may or may not reload the listed databases.
SIGILL	Dumps statistics data into **named.stats**. Statistics data is appended to the file.
SIGINT	The **named8** daemon dumps the current database to a file named **/var/tmp/named_dump.db**. In the dump file, names with the label **name error** indicate negative cache entries. This happens when a server responds that the specified domain name does not exist. Names labeled as **data error** also indicate negative cache entries. This happens when a server responds that there are no records of the specified type for the (valid) domain name.
SIGUSR1	The **named8** daemon turns on debugging; each subsequent **SIGUSR1** signal increments the debugging level. The debugging information is written to the **/var/tmp/named.run** file.
SIGUSR2	The **named8** daemon turns off debugging.

Examples

1. To start the **named8** daemon normally, enter the following:

   ```
   startsrc -s named
   ```

   This command starts the daemon. You can use this command in the **rc.tcpip** file or on the command line. The **-s** flag specifies that the subsystem that follows is to be started. The process ID of the **named8** daemon is stored in the **/etc/named.pid** file upon startup.
2. To stop the **named8** daemon normally, enter:

   ```
   stopsrc -s named
   ```

   This command stops the daemon. The **-s** flag specifies that the subsystem that follows is to be stopped.
3. To get short status from the **named8** daemon, enter:

   ```
   lssrc -s named
   ```

This command returns the name of the daemon, the process ID of the daemon, and the state of the daemon (active or inactive).

4. To enable debugging for the **named8** daemon, enter:

   ```
   traceson -s named
   ```

   OR

   ```
   kill -30 `cat /etc/named.pid`
   ```

   The **named8** daemon turns on debugging in response to either of these commands; each subsequent command increments the debugging level. The debugging information is written to the **/var/tmp/named.run** file.

5. To turn off debugging for the **named8** daemon, enter:

   ```
   tracesoff
   ```

   OR

   ```
   kill -35 `cat /etc/named.pid`
   ```

   Either of these commands immediately turns off all debugging.

6. To start the **named8** daemon at the highest debugging level using the **startsrc** command, enter the following:

   ```
   startsrc -s named -a -d11
   ```

   This command writes debugging messages to the **/var/tmp/named.run** file.

Files

/usr/sbin/named8	Contains the **named8** daemon.
/usr/sbin/named8-xfer	Provides the functionality of the slave name server's inbound zone transfer.
/etc/named.conf	Specifies the configuration of the **named8** daemon including some basic behaviors, logging options, and locations of the local databases.
/etc/resolv.conf	Specifies the use of domain name services.
/etc/rc.tcpip	Initializes daemons at each system restart.
/etc/named.pid	Stores process ID.
/etc/services	Defines socket service assignments.
/usr/samples/tcpip/named.conf	Contains the sample **named.conf** file with directions for its use.
/usr/samples/tcpip/named.data	Contains the sample DOMAIN data file with directions for its use.
/usr/samples/tcpip/hosts.awk	Contains the sample **awk** script for converting an **/etc/hosts** file to an **/etc/named.data** file. This file also contains directions for its use.
/usr/samples/tcpip/addrs.awk	Contains the sample **awk** script for converting an **/etc/hosts** file to an **/etc/named.rev** file. This file also contains directions for its use.

Related Information

The **nslookup** command, **traceroute** command, **kill** command, **chroot** command.

The **named4** daemon.

The **named.conf** file format, **DOMAIN Cache** file format, **DOMAIN Data** file format, **DOMAIN Reverse Data** file format, **DOMAIN Local Data** file format, **resolv.conf** file format.

TCP/IP Name Resolution and TCP/IP Daemons in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

Configuring Name Servers and Planning for DOMAIN Name Resolution in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

named9 Daemon

Purpose

Internet domain name server.

Syntax

named9 [**-c** *config-file*] [**-d** *debug-level*] [**-f**] [**-g**] [**-n** *#cpus*] [**-p** *port*] [**-s**] [**-v**] [**-x** *cache-file*]

Description

named9 is a Domain Name System (DNS) server, part of the BIND 9 distribution from ISC. For more information on the DNS, see RFCs 1033, 1034, and 1035. When invoked without arguments, named will read the default configuration file **/etc/named.conf**, read any initial data, and listen for queries.

Flags

-c*config-file*	Use *config-file* as the configuration file instead of the default, /etc/named.conf. To ensure that reloading the configuration file continues to work after the server has changed its working directory due to to a possible directory option in the configuration file, config-file should be an absolute pathname.
-d*debug-level*	Set the daemon's debug level to *debug-level*. Debugging traces from named become more verbose as the debug level increases.
-f	Run the server in the foreground (i.e. do not daemonize).
-g	Run the server in the foreground and force all logging to stderr.
-n *#cpus*	Create *#cpus* worker threads to take advantage of multiple CPUs. If not specified, named will try to determine the number of CPUs present and create one thread per CPU. If it is unable to determine the number of CPUs, a single worker thread will be created.
-p*port*	Listen for queries on port *port*. If not specified, the default is port 53.
-s	Write memory usage statistics to **stdout** on exit. **Note:** This option is mainly of interest to BIND 9 developers and may be removed or changed in a future release.
-v	Report the version number and exit.
-x *cache-file*	Load data from cache-file into the cache of the default view. **Attention:** This option must not be used. It is only of interest to BIND 9 developers.

Signals

In routine operation, signals should not be used to control the nameserver; **rndc** should be used instead.

SIGHUP	Forces a reload of the server.
SIGINT, SIGTERM	Shut down the server.

The result of sending any other signals to the server is undefined.

Configuration

A complete description of the **named9** configuration file is provided in the BIND 9 Administrator Reference Manual.

Files

/usr/sbin/named9	Contains the **named9** daemon.
/etc/named.conf	The default configuration file.
/etc/named.pid	The default process-id file.

Related Information

The **named4**, and **named8** daemons.

The **named.conf** file format,

RFC 1033, RFC 1034, RFC 1035, rndc(8), and lwresd(8).

The BIND 9 Administrator Reference Manual.

---

namerslv Command

Purpose

Directly manipulates domain name server entries for local resolver routines in the system configuration database.

Syntax

To Add a Name Server Entry

namerslv -a { **-i** *IPAddress* | **-D** *DomainName*| **-S** *SearchList*}

To Delete a Name Server Entry

namerslv -d { **-i** *IPAddress* | **-n** | **-l**}

To Delete All Name Server Entries

namerslv -X [**-l**]

To Change a Name Server Entry

namerslv -c *DomainName*

To Display a Name Server Entry

namerslv -s [**-l** | **-n** | **-l**] [**-Z**]

To Create the Configuration Database File

namerslv -b [**-i** *IPAddress* [**-D** *DomainName*] [**-S** *SearchList*]]

To Rename the Configuration Database File

namerslv -E *FileName*

To Move the Configuration Database File to Prevent Name Server Use

namerslv -e

To Import a File into the Configuration Database File

namerslv -B *FileName*

To Change a Search List Entry

namerslv -C *Search List*

Description

The **namerslv** low-level command adds or deletes domain name server entries for local resolver routines in the system configuration database. By default, the system configuration database is contained in the **/etc/resolv.conf** file.

To use a name server, do one of the following:

- Specify a file name to use as the system configuration database.
- Specify an Internet Protocol address and, optionally, a domain name.

The **namerslv** command can show one or all domain name server entries in the system configuration database. The **namerslv** command can also rename the **/etc/resolv.conf** file so that it stops using a name server.

There are three types of domain name server entries:

- A domain entry identifying the name of the local Internet domain.
- A name server entry that identifies the Internet address of a domain name server for the local domain. The address must be in dotted decimal format.
- A search list entry that lists all the domains to search when resolving hostnames. This is a space delimited list.

One domain entry and a maximum of three name server entries can exist in the system configuration database. The **MAXNS** global variable in the **/usr/include/resolv.h** file defines the maximum number of name servers. One search entry can exist.

You can use the Web-based System Manager Network application (**wsm network** fast path) to run this command. You could also use the System Management Interface Tool (SMIT) **smit namerslv** fast path to run this command.

Flags

-a	Adds an entry to the system configuration database. The **-a** flag must be used with either the **-i** or **-D** flag.
-B *FileName*	Restores the **/etc/resolv.conf** file from the file specified by the *FileName* variable.
-b	Creates the system configuration database, using the **/etc/resolv.conf.sv** file. If the **/etc/resolv.conf.sv** file does not exist, an error is returned. **Note:** The **/etc/resolv.conf.sv** file is not shipped with the system. You have to create the file before the **-b** flag will work.
-C	Changes the search list in the **/etc/resolv.conf** file.
-c *DomainName*	Changes the domain name in the system configuration database.
-D	Indicates that the command deals with the domain name entry.
-d	Deletes an entry in the system configuration database. It must be used with the **-i** *IPAddress* flag or the **-n** flag. The **-i** flag deletes a name server entry. The **-n** flag deletes the domain name entry.
-E *FileName*	Renames the system configuration database file, so you can stop using a name server. The **/etc/resolv.conf** file is moved to the file specified by the *FileName* variable.
-e	Moves the **/etc/resolv.conf** file to the **/etc/resolv.conf.sv** file, preventing use of a name server.
-I	(Uppercase i) Specifies that the **-s** flag or **-X** flag should print all name server entries.
-i *IPAddress*	Indicates that the command deals with a name server entry. Use dotted decimal format for the given IP address.
-l	(Lowercase L) Specifies that the operation is on the search list. Use this flag with the **-d** and **-s** flag.
-n	Specifies that the operation is on the domain name. Use this flag with the **-d** flag and the **-s** flag.
-S *SearchList*	Changes the search list in the system configuration database.

-s	Shows all domain and name server entries in the configuration system database. If you use the **-i** flag, the **namerslv** command shows all name server entries. If you use the **-n** flag, the **namerslv** command shows the domain name entry found in the database.
-X	Deletes all entries in the database. Use the **-I** flag with this flag to delete all name server entries.
-Z	Generates the output of the query in colon format. This flag is used when the **namerslv** command is called from the SMIT usability interface.

Examples

1. To add a domain entry with a domain name of `abc.aus.century.com`, type:

   ```
   namerslv  -a  -D abc.aus.century.com
   ```

2. To change the `abc.aus.century.com` domain entry to the domain name `xyz.aus.century.com`, type:

   ```
   namerslv xyz.aus.century.com
   ```

3. To add a name server entry with IP address `192.9.201.1`, type:

   ```
   namerslv  -a  -i 192.9.201.1
   ```

4. To show all system configuration database entries related to domain name server information used by local resolver routines, type:

   ```
   namerslv  -s
   ```

   The output is given in the following format:

   ```
   domain xyz.aus.century.com
       name server 192.9.201.1
   ```

5. To rename the **/etc/resolv.conf** file to stop using the name server and specify the new file name, `/etc/resolv.back`, type:

   ```
   namerslv  -E /etc/resolv.back
   ```

Files

/usr/sbin/namerslv	Contains the **namerslv** command.
/etc/resolv.conf	Contains the default system configuration database.
/etc/resolv.conf.sv	Contains the old system configuration database.

Related Information

The **chnamsv** command, **lsnamsv** command, **mknamsv** command, **nslookup** command, **rmnamsv** command, **traceroute** command.

Naming and TCP/IP Daemons in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.2 Web-based System Manager Administration Guide*.

The TCP/IP Reference in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

ncheck Command

Purpose

Generates path names from i-node numbers.

Syntax

ncheck [[[**-a**] [**-i** *InNumber ...*]] | [**-s**]] [*FileSystem*]

Description

The **ncheck** command displays the i-node number and path names for filesystem files. It uses question marks (??) displayed in the path to indicate a component that could not be found. Path names displayed with ... (ellipses) at the beginning indicate either a loop or a path name of greater than 10 entries. The **ncheck** command uses a simple hashing alogrithm to reconstruct the path names that it displays. Because of this, it is restricted to filesystems with less than 50,000 directory entries.

Flags

-a	Lists the . (dot) and .. (dot dot) file names.
-i *InNumber*	Lists only the file or files specified by the *InNumber* parameter.
-s	Lists only special files and files with set-user-ID mode.

Examples

1. To list the i-node number and path name of each file in the default file systems, enter:

   ```
   ncheck
   ```

2. To list all the files in a specified file system, enter:

   ```
   ncheck  -a /
   ```

   This lists the i-node number and path name of each file in the **/** (root) file system, including the .(dot) and .. (dot dot) entries in each directory.

3. To list the name of a file when you know its i-node number, enter:

   ```
   ncheck  -i 690 357 280 /tmp
   ```

   This lists the i-node number and path name for every file in the **/tmp** file system with i-node numbers of `690`, `357`, or `280`. If a file has more than one link, all of its path names are listed.

4. To list special and set-user-ID files, enter:

   ```
   ncheck  -s /
   ```

   This lists the i-node and path name for every file in the **/** (root) file system that is a special file (also called a device file) or that has set-user-ID mode enabled.

Related Information

The **fsck** command, **sort** command.

File Systems in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*.

ndp Command

Purpose

IPv6 neighbor discovery display and control.

Syntax

ndp [**-n**] *hostname*

ndp [**-n**] **-a**

ndp [**-d**] *hostname*

ndp [**-i** *interface_index*] **-s** *hostname addr* [**temp**]

Description

The **ndp** program displays and modifies the IPv6-to-Ethernet, or the IPv6-to-TokenRing address translation tables used by the IPv6 neighbor discovery protocol.

With no flags, the program displays the current **ndp** entry for *hostname*. The host may be specified by name or by number, using IPv6 textual notation.

Flags

- a	Displays all of the current **ndp** entries.
- d	Lets a super-user delete an entry for the host called *hostname* with the **-d** flag.
- i *interface_index*	Specifies the index of the interface to use when an **ndp** entry is added with the **-s** flag (useful with the local-link interface).
- n	Shows network addresses as numbers (normally **ndp** attempts to display addresses symbolically).
- s *hostname addr*	Creates an **ndp** entry for *hostname* with the Hardware address *addr*. The Hardware address is given as six hex bytes separated by colons. The entry is permanent unless the **temp** is specified in the command.

Examples

This is an example output from the **- a** flag:

```
# ndp -a
e-crankv6 (::903:9182) at link#2 0:20:af:db:b8:cf
e-crankv6-11 (fe80:0:100::20:afdb:b8cf) at link#2 0:20:af:db:b8:cf
# ndp -d e-crankv6-11
e-crankv6-11 (fe80:0:100::20:afdb:b8cf) deleted
```

Related Information

The **ifconfig** command, **ndpd-host** command, **ndpd-router** command, **autoconf6** command.

ndpd-host Daemon

Purpose

NDP daemon for an host.

Syntax

ndpd-host [**-d**] [**-v**] [**-t**]

Description

The **ndpd-host** command manages the Neighbor Discovery Protocol (NDP) for non-kernel activities: Router Discovery, Prefix Discovery, Parameter Discovery and Redirects. The **ndpd-host** command deals with the default route, including default router, default interface and default interface address.

Interfaces

The **ndpd-host** command knows about IEEE and CTI point to point interfaces. The **ndpd-host** command exchanges packets on all the known interfaces UP with a Link-Local Address. Any change of status of an interface is detected. If an interface goes down or loses its Link-Local address, the NDP processing is stopped on this interface. If an interface goes up, the NDP processing is started.

The IEEE interfaces are configured using the **autoconf6** command. The PPP interfaces are configured using the **pppd** daemon. The token negotiation defines the Link-Local addresses. In order to send Router Advertisements over a CTI configured tunnel, it must have local and distant Link-Local addresses.

> **Note:** For all the up point to point interfaces, **ndpd-host** sets a local route via lo0 for local addresses.

Flags

-d	Enables debugging (exceptional conditions and dump).
-v	Logs all interesting events (daemon.info and console).
-t	Adds a time stamp in each log.

Signals

SIGUSR1	Turns on verbose.
SIGUSR2	Turns off verbose.
SIGINT	Dumps the current state of ndpd-host to syslog or stdout.
SIGTERM	Cleans up ndpd-host and exits.

Related Information

The **ifconfig** command, **route** command, **autoconf6** command, and the **ndpd-router** command.

---

ndpd-router Daemon

Purpose

NDP and RIPng daemon for a router.

Syntax

ndpd-router [**-r**] [**-p**] [**-M**] [**-O**] [**-s**] [**-q**] [**-g**] [**-n**] [**-R**] [**-S**] [**-d**] [**-t**] [**-v**] [**-H**] [**-m**] [**-u** *port*] [**-D** *max*[*min*[**/***life*]]] [**-P** [*invlife*]**/**[*deplife*]] [**-T** [*reachtim*]**/**[*retrans*]**/**[*hlim*]]

Description

The **ndpd-router** daemon manages the Neighbor Discovery Protocol (NDP) for non-kernel activities. It receives Router Solicitations and sends Router Advertisements. It can also exchange routing information using the RIPng protocol.

The **/etc/gateway6** file provides options for **ndpd-router**. This file can be modified while the program is running. The changes are checked before any emission or reception of message, or on reception of the

HUP signal. The file contains directives, one by line (with # as comment). All the IPv6 addresses and prefixes in the file must be in numeric form. No symbolic name is allowed. Except for the gateway directive, each line begins with a keyword and is made of options of the form *key=argument*.

Interfaces

The **ndpd-router** daemon knows about IEEE and CTI point to point interfaces. The **ndpd-router** daemon exchanges packets on all the known interfaces UP with a Link-Local Address. Any change of status of an interface is detected. If an interface goes down or loses its Link-Local address, the NDP and RIPng processing is stopped on this interface. If an interface goes up, the NDP and RIPng processing is started.

To send Router Advertisements or RIPng packets or both, local *and* remote Link-Local addresses must be configured.

Flags

-H	Enables the system to process NDP features needed to function as a mobile IPv6 home agent
-m	Enables the system to aid movement detection for mobile IPv6 mobile nodes.
-D *max* [*min*[**/***life*]]	Sends Unsolicited Router Advertisements at intervals from *min* to *max* seconds. Default *max* value is 600 seconds, valid range is 4 to 1800 seconds. Default *min* equals to *max* / 3, valid range is from 1 to 0.75 * *max*. The router lifetime is set with *life*, default value is 10 * max. Valid range is 0 to 65535 seconds.
-T [*reachtim*] **/** [*retrans*] **/** [*hlim*]	Sets the BaseReachableTime field to *reachim* seconds, if *reachim* is not zero. If *retrans* is not zero, sets the RetransTime field to *retrans* seconds. If *hlim* is not zero, sets the hop limit field in Router Advertisements to *hlim*.
-M	Sets the **M** flag (stateful configuration) in advertisements.
-O	Sets the **O** flag (other stateful information) in advertisements
-p	Does not offer prefixes (learned from interface configuration).
-P [invlife]/[deplife]	Sets the invalid life value and the deprecated life value for announced prefixes (in seconds). The default value is 0xffffffff (infinite).
-r	Does not offer to be the default router in Router Advertisements.
-s	Enables the RIPng protocol (the default is: RIPng disabled).
-q	Enables the RIPng protocol, but does not send RIPng packets.
-g	Broadcast a default route in RIPng.
-n	Does not install routes received by RIPng.
-u *port*	Uses UDP port *port* for RIPng. The default is 521.
-R	Uses split horizon without corrupting reverse for RIPng.
-S	Does not use any split horizon for RIPng.
-d	Enables debugging (exceptional conditions and dump).
-v	Logs all interesting events (daemon.info and console).
-t	Adds timestamps in logged messages.

Available directives

The main directives for the **/etc/gateway6** file are:

option [*option-directive* **...]**
Sets per-interface/default options.

prefix [*prefix-directive* **...]**
Sets per-interface/default prefix processing options.

filter [*filter-directive* **...]**
Sets per-interface/default filters.

gateway directives
Sets routes in RIPng packets or in the kernel.

Each of these directives is explained in more detail below.

The option directive

Sets different per-interface options.

Any value settings for the **option** directive which follow the **if** option must appear in a comma-separated list.

Note: At least one option (other than the **if** option) must be specified following the **option** directive. If the **if** option is specified, it must be the first option following the **option** directive. There must be a space between the **if** option and any comma-separated list of options which follow.

Syntax:

option [**if**=*n1*,*n2*] **ripin**=(**y**|**n**),**ripout**=(**y**|**n**|**S**|**R**),**rtadv**=(**y**|**n**|*min*[/*max*]),**flag**=[**M**|**O**],**life**=*Seconds*,**reach**=*Seconds*,**retrans**=*Seconds*

if=*list* **interface=***list*	If there is no keyword, the option directive is a default option. If there is an interface field, the option parameters apply only to the listed interfaces. The list is comma-separated. You can use `le*` to match all the leX interfaces. The default option must be the first line in the **/etc/gateway6** file.
mtu [**=***mtuval*]	Advertises a MTU value of *mtuval* in router advertisements. If there is no *mtuval* argument, the advertised MTU is the MTU of the interface. If *mtuval* is 0, suppress the advertisement of MTU.
ripin=(**n**|**y**)	Does not listen (listen) to incoming RIPng packets. Does not send (send) RIPng packets. With the **-S** flag, do not use split horizon. With the **-R** flag, use split horizon without poisoning reverse.
rtadv=(**n**|**y**|*min* [**/***max*])	Does not send (send) router advertisements. With *min*[**/***max*] option, set the interval (in seconds) between router advertisements.
flag={**M**|**O**}	Sets the stateful mode flags in router advertisements. **M** Uses stateful configuration **O** Uses stateful configuration, but not for addresses
life=*Seconds*	Sets the router life field in router advertisements (in seconds).
reach=*Seconds*	Sets the reachable field in router advertisements (in seconds).
retrans=*Seconds*	Sets the retransmit interval field in router advertisements (in seconds).

The prefix directive

Defines the prefixes announced in Router advertisement directives. If there is no prefix-directive for an interface, the router advertisement contains the list of prefixes deduced from the address list of the interface. If there are prefix-directives, the router advertisement contains the list of prefixes defined by the different prefix directives (in order). No prefix is installed in the kernel. If there is one directive of the form prefix *prefix*=*none*, no prefix list is advertised.

Syntax:

prefix if=n **prefix**=(**none**|*xxx*::/*PrefixLength*) **flag**=[**L**][**A**] **valid**=*Seconds* **deprec**=*Seconds*

if=*Interface* or **interface**=*Interface*	Specifies the interface on which the directive applies. The **if** keyword is mandatory for the **prefix** directive. It is not an option.
prefix=*xxx*::/*PrefixLength*	The advertised prefix.
flag=[**L**][**A**]	Set the **L** and/or **A** flag for the prefix (the default is **LA**).
deprec=*Seconds*	Set the deprecated time (in seconds) for the prefix.
valid=*Seconds*	Set the validity time (in seconds) for the prefix.

The filter directive

Define a filter pattern for incoming (**filter**=**in**) or outgoing (**filter**=**out**) RIPng packets. There is one incoming and one outgoing filter per interface, and one default incoming and one default outgoing filter for interfaces without explicit filter.

Any received RIPng information is tested against the input filter of the interface, or, if there is none, against the default input filter. The static interface routes are seen as input information coming from the interface and from a gateway with the link local address of the interface. The routes set by a gateway directive with a **gateway** keyword are seen as input information coming from the specified interface and gateway. The default route (**-g** flag) and the routes set by a gateway directive without a **gateway** keyword are seen as input information coming from gateway :: and no interface (the default input filter applies).

Any sent RIPng information is tested against the output filter of the interface, or, if there is none, against the default output filter.

Each filter is a sequence of matching patterns. The patterns are tested in order. Each pattern can test the prefix length, the source gateway (for input filters and that the prefix (padded with zeroes) matches a fixed prefix. If a pattern contains more than one test description, the match is the conjunction of all the tests. The first matching pattern defines the action to perform. If no pattern matches, the default action is accept. The possible actions are accept, reject and truncate/*NumberOfBits*. The truncate/*NumberOfBits* action means: if the pattern matches and if prefix length is greater or equal to *NumberOfBits*, accept the prefix with new length *NumberOfBits*. The accepted prefix is immediately accepted, that is, not checked again against the filters.

For example, the following directive inhibits sending host routes on any interface without an explicit outgoing filter:

```
filter=out length==128 action=reject
```

Syntax:

filter=(**in**|**out**) [**if**=*n1*,*n2*] **prefix**=*xx*::/*NumberOfBits* **gateway**=*xxx* **length**=(=|>=|<=|<|>)*NumberOfBits* **action**=(**accept**|**reject**|**truncate**/*xx*)

if=*list* or **interface**=*list*	If there is no interface keyword, the filter directive is a default option. If there is an interface field, the filter pattern is added at the end of the filters of all specified interfaces. The list is comma-separated. For example, you can specify **interface**=le* to specify all the leX interfaces.
prefix=*xxx*::/*NumberOfBits*	The pattern matches only if *xxx*::/*NumberOfBits* is a prefix of the prefix in the RIPng packet.
gateway=*xxx*	The pattern matches only if the RIPng message comes from source address *xxx*, only in incoming filters.

length=(=|>=|<=|<|>)*NumberOfBits*	The pattern match only if the prefix length in the RIPng message is equal to (or greater than, less than, etc., depending on the operator specified) to *NumberOfBits*.
action=(**accept**|**reject**|**truncate**/*NumberOfBits*)	Specify the action to perform if the pattern matches: accept the message, reject the message, accept but truncate the prefix to *NumberOfBits* bits.

Gateway directives

The gateway directives allow the user to set up routes in RIPng packets and/or in the kernel. These directives must appear at the end of the **/etc/gateway6** file, after the other directives.

Syntax:

xxx::/*NumberOfBits* **metric** *Value*

xxx::/*NumberOfBits* **metric** *Value* **gateway** *IPv6Address ifname*

The second syntax is used to add the route to the kernel.

Examples

The following examples are of the **/etc/gateway6** file.

On a site where all addresses are of the form `5f06:2200:c001:0200:xxxx`, the following example means that only one route, describing all the site, is exported on all the Configured Tunnel Interface (CTI) **ctiX** interfaces. The keyword abbreviations shown are valid.

```
filt=out if=cti* pref=5f06:2200:c001:0200::/64 len=>=64 act=trunc/64
```

Setting a default outgoing route:

```
::/0 metric 2 gateway 5f06:2200:c102:0200::1 cti0
```

Declare that any CTI interface active with RIPng defines a default route:

```
filter=in if=cti* act=trunc/0
```

The following example defines a site with an exterior connection cti0, which aggregates other sites connected through ctiX, and which uses split horizon without poisoned reverse. The order of the lines is important, as all filter descriptions apply to cti0.

```
option if=cti* ripout=R
filter=out if=cti0 prefix=5f06:2200::/24 len=>=24 act=trunc/24
filt=out if=cti* pref=5f06:2200:c001:0200::/64 len=>=64 act=trunc/64
filter=in if=cti0 act=trunc/0
filter=in if=cti* prefix=5f06:2200::/24 len=>=24 act=trunc/64
filter=in if=cti* act=reject
```

Diagnostics

All errors are logged at the **daemon.err** level, unless the debug option is set. This includes all the syntax errors in the **/etc/gateway6** file and configuration mismatches between different routers.

Signals

ndpd-router responds to the following signals:

SIGINT	Dumps its current state to syslog, if syslog is defined. Otherwise, dumped to stdout.
SIGHUP	The **/etc/gateway6** file is read again.
SIGUSR1	Verbosity is incremented.
SIGUSR2	Verbosity is reset.

SIGTERM Resets to a resonable state and stops.
SIGQUIT Resets to a resonable state and stops.

Files

/etc/gateway6

Related Information

The **ifconfig** command, **kmodctrl** command, **mobip6reqd** command, **mobip6ctrl** command, **rc.mobip6** command, **route** command, **autoconf6** command, **ndpd-host** command.

The Mobile IPv6 section in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

---

ndx Command

Purpose

Creates a subject-page index for a document.

Syntax

ndx [*SubjectFile*] *" FormatterCommandLine "*

Description

The **ndx** command, given a list of subjects (*SubjectFile*), searches a specified English-language document and writes a subject-page index to standard output.

The document must include formatting directives for the **mm**, **mmt**, **nroff**, or **troff** commands. The formatter command line informs the **ndx** command whether the **troff** command, **nroff** command, **mm** command, or **mmt** command can be used to produce the final version of the document. These commands do the following:

troff or **mmt** Specifies the **troff** command as the formatting program.
nroff or **mm** Specifies the **nroff** command as the formatting program.

Parameters

SubjectFile Specifies the list of subjects that are included in the index. Each subject must begin on a new line and have the following format:

word1[*word2*...][,*wordk*...]

For example:

```
printed circuit boards
arrays
arrays, dynamic storage
Smith, W.P.
printed circuit boards, channel-oriented
                                multi-layer
Aranoff
University of Illinois
PL/1
```

The subject must start in column one.

FormatterCommandLine

Creates the final form of the document. The syntax for this parameter is as follows:

Formatter [*Flag...*] *File...*

mm -Tlp File(s)
nroff -mm -Tlp -rW60 File(s)
troff -rB2 -Tibm3816 -r01.5i File(s)

For more information on the formatter command line, see the **mm** command, **mmt** command, **nroff** command, and **troff** command.

The flags specified by the *Flag* variable are those that are given to the **troff**, **nroff**, **mm**, or **mmt** command in printing the final form of the document. These flags are necessary to determine the correct page numbers for subjects as they are located in the document. The **ndx** command does not cause the final version of the document to be printed. The author must create the document separately. Use the indexer only after the document is complete and cannot undergo further changes.

Examples

1. The following command produces a subject-page index for the `file` document and takes its subjects from the `subfile` list:

   ```
   ndx subfile "nroff -mm -rW70 file" > indexfile
   ```

   The page numbers correspond to the document produced by:

   ```
   nroff -mm -rW70 file
   ```

2. The following command produces a subject-page index for the documents `ch1`, `ch2`, and `ch3`:

   ```
   ndx subfile "mm -rW60 -rN2 -rO0 ch1 ch2 ch3" > indexfile
   ```

   The page numbers would correspond to the documents produced by:

   ```
   mm -rW60 -rN2 -rO0 ch1 ch2 ch3
   ```

3. The following command produces a subject-page index for the document `file`:

   ```
   ndx Subjfile "troff -rB2 -rW5i -rO1.5i -mm file" > indexfile
   ```

   The page numbers correspond to the document produced by entering:

   ```
   troff -rB2 -rW5i -rO1.5i -mm file
   ```

Related Information

The **mm** command, **mmt** command, **nroff** command, **subj** command, **troff** command.

---

neqn Command

Purpose

Formats mathematical text for the **nroff** command.

Syntax

neqn [**-d** *Delimiter1Delimiter2*] [**-f** *Font*] [**-p** *Number*] [**-s** *Size*] [**—**] [*File* ... | **-**]

Description

The **neqn** command is an **nroff** preprocessor for formatting mathematical text on typewriter-like terminals. Pipe the output of the **neqn** command into the **nroff** command as follows:

neqn [*Flag...*] *File...* **|** **nroff** [*Flag...*] **|** [*Printer*]

The **neqn** command reads one or more files. If no files are specified for the *File* parameter or the **-** (minus sign) flag is specified as the last parameter, standard input is read by default. A line beginning with the **.EQ** macro marks the start of equation text. The end of equation text is marked by a line beginning with the **.EN** macro. These lines are not altered by the **nroff** command, so they can be defined in macro packages to provide additional formatting functions such as centering and numbering.

The **—** (double dash) delimiter indicates the end of flags.

Depending on the target output devices, **neqn** command output formatted by the **nroff** command may need to be post-processed by the **col** command to produce correct output.

The **eqn** command gives more information about the input format and keywords used.

Flags

-d*Delimiter1Delimiter2*	Sets two ASCII characters, *Delimiter1* and *Delimiter2,* as delimiters of the text to be processed by the **neqn** command, in addition to input enclosed by the **.EQ** and **.EN** macros. The text between these delimiters is treated as input to the **neqn** command. Within a file, you can also set delimiters for **neqn** text using the **delim** *Delimiter1Delimiter2* request. These delimiters are turned off by the **delim off** request. All text that is not between delimiters or the **.EQ** macro and **.EN** macro is passed through unprocessed.
-f*Font*	Changes font in all the **neqn** command-processed text to the value specified by the *Font* variable. The *Font* value (a font name or position) must be one or two ASCII characters.
-p*Number*	Reduces subscripts and superscripts to the specified number of points in size. The default is 3 points.
-s*Size*	Changes point size in all the **neqn** command-processed text to the value specified by the *Size* variable.
-	Reads from standard input.
—	(double dash) Marks the end of the flags.

Files

/usr/share/lib/pub/eqnchar	Contains special character definitions.

Related Information

The **checkeq** command, **col** command, **eqn** command, **mm** command, **nroff** command, **tbl** command.

The **.EN** macro, **.EQ** macro, **mm** macro.

The **eqnchar** file format.

---

netpmon Command

Purpose

Monitors activity and reports statistics on network I/O and network-related CPU usage.

Syntax

netpmon [**-o** *File*] [**-d**] [**-T** *n*] [**-P**] [**-t**] [**-v**] [**-O** *ReportType ...*] [**-i** *Trace_File* **-n** *Gennames_File*]

Description

The **netpmon** command monitors a trace of system events, and reports on network activity and performance during the monitored interval. By default, the **netpmon** command runs in the background while one or more application programs or system commands are being executed and monitored. The **netpmon** command automatically starts and monitors a trace of network-related system events in real time. By default, the trace is started immediately; optionally, tracing may be deferred until the user issues a **trcon** command. When tracing is stopped by a **trcstop** command, the **netpmon** command generates all specified reports and exits.

The **netpmon** command can also work in offline mode, that is, on a previously generated trace file. In this mode, a file generated by the **gennames** command is also required. The gennames file should be generated immediately after the trace has been stopped, and on the same machine. When running in offline mode, the **netpmon** command cannot recognize protocols used by sockets, which limits the level of detail available in the socket reports.

The **netpmon** command reports on the following system activities:

Note: The **netpmon** command does not work with NFS3(ONC+)

CPU Usage
The **netpmon** command monitors CPU usage by all threads and interrupt handlers. It estimates how much of this usage is due to network-related activities.

Network Device-Driver I/O
The **netpmon** command monitors I/O operations through Micro-Channel Ethernet, token-ring, and Fiber-Distributed Data Interface (FDDI) network device drivers. In the case of transmission I/O, the command also monitors utilizations, queue lengths, and destination hosts. For receive ID, the command also monitors time in the demux layer.

Internet Socket Calls
The **netpmon** command monitors all **send**, **recv**, **sendto**, **recvfrom**, **read**, and **write** subroutines on Internet sockets. It reports statistics on a per-process basis, for each of the following protocol types:

- Internet Control Message Protocol (ICMP)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

NFS I/O
The **netpmon** command monitors **read** and **write** subroutines on client Network File System (NFS) files, client NFS remote procedure call (RPC) requests, and NFS server read or write requests. The command reports subroutine statistics on a per-process or optional per-thread basis and on a per-file basis for each server. The **netpmon** command reports client RPC statistics for each server, and server read and write statistics for each client.

Any combination of the preceding report types can be specified with the command line flags. By default, all the reports are produced.

Notes: The reports produced by the **netpmon** command can be quite long. Consequently, the **-o** flag should usually be used to write the report to an output file. The **netpmon** command obtains performance data using the system trace facility. The trace facility only supports one output stream. Consequently, only one **netpmon** or **trace** process can be active at a time. If another **netpmon** or **trace** process is already running, the **netpmon** command responds with the message:

```
/dev/systrace: Device busy
```

While monitoring very network-intensive applications, the **netpmon** command may not be able to consume trace events as fast as they are produced in real time. When that happens, the error message:

```
Trace kernel buffers overflowed, N missed entries
```

displays on standard error, indicating how many trace events were lost while the trace buffers were full. The **netpmon** command continues monitoring network activity, but the accuracy of the report diminishes by some unknown degree. One way to avoid overflow is to increase the trace buffer size using the **-T** flag, to accommodate larger bursts of trace events before overflow. Another way to avoid overflow problems all together is to run netpmon in offline mode.

When running in memory-constrained environments (where demand for memory exceeds supply), the **-P** flag can be used to pin the text and data pages of the real-time **netpmon** process in memory so the pages cannot be swapped out. If the **-P** flag is not used, allowing the **netpmon** process to be swapped out, the progress of the **netpmon** command may be delayed such that it cannot process trace events fast enough to prevent trace buffer overflow.

If the **/unix** file and the running kernel are not the same, the kernel addresses will be incorrect, causing the **netpmon** command to exit.

Flags

-d Starts the **netpmon** command, but defers tracing until the **trcon** command has been executed by the user. By default, tracing is started immediately.

-i *Trace_File* Reads trace records from the file *Trace_File* produced with the **trace** command instead of a live system. The trace file must be rewritten first in raw format using the **trcpt -r** command. This flag cannot be used without the **-n** flag.

-n *Gennames_File* Reads necessary mapping information from the file *Gennames_File* produced by the **gennames** command. This flag is mandatory when the **-i** flag is used.

-o *File* Writes the reports to the specified *File*, instead of to standard output.

-O *ReportType ...* Produces the specified report types. Valid report type values are:

- **cpu** CPU usage
- **dd** Network device-driver I/O
- **so** Internet socket call I/O
- **nfs** NFS I/O
- **all** All reports are produced. This is the default value.

-P Pins monitor process in memory. This flag causes the **netpmon** text and data pages to be pinned in memory for the duration of the monitoring period. This flag can be used to ensure that the real-time **netpmon** process does not run out of memory space when running in a memory-constrained environment.

-t Prints CPU reports on a per-thread basis.

-T *n* Sets the kernel's trace buffer size to *n* bytes. The default size is 64000 bytes. The buffer size can be increased to accommodate larger bursts of events, if any. (A typical event record size is on the order of 30 bytes.)

> **Note:** The trace driver in the kernel uses double buffering, so actually two buffers of size *n* bytes will be allocated. These buffers are pinned in memory, so they are not subject to paging.

-v Prints extra information in the report. All processes and all accessed remote files are included in the report instead of only the 20 most active processes and files.

Reports

The reports generated by the **netpmon** command begin with a header, which identifies the date, the machine ID, and the length of the monitoring period in seconds. This is followed by a set of summary and detailed reports for all specified report types.

CPU Usage Reports

Process CPU Usage Statistics: Each row describes the CPU usage associated with a process. Unless the verbose option is specified, only the 20 most active processes are listed. At the bottom of the report, CPU usage for all processes is totaled, and CPU idle time is reported.

Process
Process name

PID Process ID number

CPU Time
Total amount of CPU time used by this process

CPU % CPU usage for this process as a percentage of total time

Network CPU %
Percentage of total time that this process spent executing network-related code

Thread CPU Usage Statistics
If the **-t** flag is used, each process row described above is immediately followed by rows describing the CPU usage of each thread owned by that process. The fields in these rows are identical to those for the process, except for the name field. (Threads are not named.)

First-Level Interrupt Handler Usage Statistics: Each row describes the CPU usage associated with a first-level interrupt handler (FLIH). At the bottom of the report, CPU usage for all FLIHs is totaled.

FLIH First-level interrupt handler description

CPU Time
Total amount of CPU time used by this FLIH

CPU % CPU usage for this interrupt handler as a percentage of total time

Network CPU %
Percentage of total time that this interrupt handler executed on behalf of network-related events

Second-Level Interrupt Handler Usage Statistics: Each row describes the CPU usage associated with a second-level interrupt handler (SLIH). At the bottom of the report, CPU usage for all SLIHs is totaled.

SLIH Second-level interrupt handler description

CPU Time
Total amount of CPU time used by this SLIH

CPU % CPU usage for this interrupt handler as a percentage of total time

Network CPU %
Percentage of total time that this interrupt handler executed on behalf of network-related events

Summary Network Device-Driver Reports

Network Device-Driver Statistics (by Device): Each row describes the statistics associated with a network device.

Device
Path name of special file associated with device

Xmit Pkts/s
Packets per second transmitted through this device

Xmit Bytes/s
Bytes per second transmitted through this device

Xmit Util
Busy time for this device, as a percent of total time

Xmit Qlen
Number of requests waiting to be transmitted through this device, averaged over time, including any transaction currently being transmitted

Recv Pkts/s
Packets per second received through this device

Recv Bytes/s
Bytes per second received through this device

Recv Demux
Time spent in demux layer as a fraction of total time

Network Device-Driver Transmit Statistics (by Destination Host): Each row describes the amount of transmit traffic associated with a particular destination host, at the device-driver level.

Host Destination host name. An * (asterisk) is used for transmissions for which no host name can be determined.

Pkts/s
Packets per second transmitted to this host

Xmit Bytes/s
Bytes per second transmitted to this host

Summary Internet Socket Reports

- *On-line mode*: **Socket Call Statistics for Each Internet Protocol (by Process)**: Each row describes the amount of **read/write** subroutine activity on sockets of this protocol type associated with a particular process. Unless the verbose option is specified, only the top 20 processes are listed. At the bottom of the report, all socket calls for this protocol are totaled.
- *Off-line mode*: **Socket Call Statistics for Each Process**: Each row describes the amount of **read/write** subroutine activity on sockets associated with a particular process. Unless the verbose option is specified, only the top 20 processes are listed. At the bottom of the report, all socket calls are totaled.

Process
Process name

PID Process ID number

Read Calls/s
Number of **read** , **recv** , and **recvfrom** subroutines per second made by this process on sockets of this type

Read Bytes/s
Bytes per second requested by the above calls

Write Calls/s
Number of **write** , **send** , and **sendto** subroutines per second made by this process on sockets of this type

Write Bytes/s
Bytes per second written by this process to sockets of this protocol type

Summary NFS Reports

NFS Client Statistics for Each Server (by File): Each row describes the amount of **read**/**write** subroutine activity associated with a file mounted remotely from this server. Unless the verbose option is specified, only the top 20 files are listed. At the bottom of the report, calls for all files on this server are totaled.

File Simple file name

Read Calls/s
Number of **read** subroutines per second on this file

Read Bytes/s
Bytes per second requested by the above calls

Write Calls/s
Number of **write** subroutines per second on this file

Write Bytes/s
Bytes per second written to this file

NFS Client RPC Statistics (by Server): Each row describes the number of NFS remote procedure calls being made by this client to a particular NFS server. At the bottom of the report, calls for all servers are totaled.

Server
Host name of server. An * (asterisk) is used for RPC calls for which no hostname could be determined.

Calls/s
Number of NFS RPC calls per second being made to this server.

NFS Client Statistics (by Process): Each row describes the amount of NFS **read**/**write** subroutine activity associated with a particular process. Unless the verbose option is specified, only the top 20 processes are listed. At the bottom of the report, calls for all processes are totaled.

Process
Process name

PID Process ID number

Read Calls/s
Number of NFS **read** subroutines per second made by this process

Read Bytes/s
Bytes per second requested by the above calls

Write Calls/s
Number of NFS **write** subroutines per second made by this process

Write Bytes/s
Bytes per second written to NFS mounted files by this process

NFS Server Statistics (by Client): Each row describes the amount of NFS activity handled by this server on behalf of particular client. At the bottom of the report, calls for all clients are totaled.

Client
Host name of client

Read Calls/s
Number of remote read requests per second processed on behalf of this client

Read Bytes/s
Bytes per second requested by this client's read calls

Write Calls/s
Number of remote write requests per second processed on behalf of this client

Write Bytes/s
Bytes per second written by this client

Other Calls/s
Number of other remote requests per second processed on behalf of this client

Detailed Reports

Detailed reports are generated for any of the specified report types. For these report types, a detailed report is produced for most of the summary reports. The detailed reports contain an entry for each entry in the summary reports with statistics for each type of transaction associated with the entry.

Transaction statistics consist of a count of the number of transactions of that type, followed by response time and size distribution data (where applicable). The distribution data consists of average, minimum, and maximum values, as well as standard deviations. Roughly two-thirds of the values are between `average - standard deviation` and `average + standard deviation`. Sizes are reported in bytes. Response times are reported in milliseconds.

Detailed Second Level Interrupt Handler CPU Usage Statistics:

SLIH Name of second-level interrupt handler

Count Number of interrupts of this type

CPU Time (Msec)
CPU usage statistics for handling interrupts of this type

Detailed Network Device-Driver Statistics (by Device):

Device
Path name of special file associated with device

Recv Packets
Number of packets received through this device

Recv Sizes (Bytes)
Size statistics for received packets

Recv Times (msec)
Response time statistics for processing received packets

Xmit Packets
Number of packets transmitted to this host

Demux Times (msec)
Time statistics for processing received packets in the demux layer

Xmit Sizes (Bytes)
Size statistics for transmitted packets

Xmit Times (Msec)
Response time statistics for processing transmitted packets

Detailed Network Device-Driver Transmit Statistics (by Host):

Host Destination host name

Xmit Packets
Number of packets transmitted through this device

Xmit Sizes (Bytes)
Size statistics for transmitted packets

Xmit Times (Msec)
Response time statistics for processing transmitted packets

Detailed Socket Call Statistics for Each Internet Protocol (by Process): (*on-line* mode)
Detailed Socket Call Statistics for Each Process: (*off-line* mode)

Process
Process name

PID Process ID number

Reads Number of **read** , **recv** , **recvfrom** , and **recvmsg** subroutines made by this process on sockets of this type

Read Sizes (Bytes)
Size statistics for **read** calls

Read Times (Msec)
Response time statistics for **read** calls

Writes
Number of **write** , **send** , **sendto** , and **sendmsg** subroutines made by this process on sockets of this type

Write Sizes (Bytes)
Size statistics for **write** calls

Write Times (Msec)
Response time statistics for **write** calls

Detailed NFS Client Statistics for Each Server (by File):

File File path name

Reads Number of NFS **read** subroutines for this file

Read Sizes (Bytes)
Size statistics for **read** calls

Read Times (Msec)
Response time statistics for **read** calls

Writes
Number of NFS **write** subroutines for this file

Write Sizes (Bytes)
Size statistics for **write** calls

Write Times (Msec)
Response time statistics for **write** calls

Detailed NFS Client RPC Statistics (by Server):

Server
Server host name

Calls Number of NFS client RPC calls made to this server

Call Times (Msec)
Response time statistics for RPC calls

Detailed NFS Client Statistics (by Process):

Process
Process name

PID Process ID number

Reads Number of NFS **read** subroutines made by this process

Read Sizes (Bytes)
Size statistics for **read** calls

Read Times (Msec)
Response time statistics for **read** calls

Writes
Number of NFS **write** subroutines made by this process

Write Sizes (Bytes)
Size statistics for **write** calls

Write Times (Msec)
Response time statistics for **write** calls

Detailed NFS Server Statistics (by Client):

Client
Client host name

Reads Number of NFS read requests received from this client

Read Sizes (Bytes)
Size statistics for read requests

Read Times (Msec)
Response time statistics for read requests

Writes
Number of NFS write requests received from this client

Write Sizes (Bytes)
Size statistics for write requests

Write Times (Msec)
Response time statistics for write requests

Other Calls
Number of other NFS requests received from this client

Other Times (Msec)
Response time statistics for other requests

Examples

1. To monitor network activity during the execution of certain application programs and generate all report types, type:

   ```
   netpmon
   <run application programs and commands here>
   trcstop
   ```

   The **netpmon** command automatically starts the system trace and puts itself in the background. Application programs and system commands can be run at this time. After the **trcstop** command is issued, all reports are displayed on standard output.
2. To generate CPU and NFS report types and write the reports to the `nmon.out` file, type:

```
netpmon -o nmon.out -O cpu,nfs
<run application programs and commands here>
trcstop
```

The **netpmon** command immediately starts the system trace. After the **trcstop** command is issued, the I/O activity report is written to the `nmon.out` file. Only the CPU and NFS reports will be generated.

3. To generate all report types and write verbose output to the `nmon.out` file, type:

```
netpmon -v -o nmon.out
<run application programs and commands here>
trcstop
```

With the verbose output, the **netpmon** command indicates the steps it is taking to start up the trace. The summary and detailed reports include all files and processes, instead of just the 20 most active files and processes.

4. To use the **netpmon** command in offline mode, type:

```
trace -a
run application programs and commands here
trcoff
gennames > gen.out
trcstop
trcrpt -r /var/adm/ras/trcfile > tracefile.r
netpmon -i tracefile.r -n gen.out -o netpmon.out
```

Related Information

The **trcstop** command, **trace** command, and **gennames** command.

The **recv** subroutine, **recvfrom** subroutine, **send** subroutine, **sendto** subroutine, and **trcoff** subroutine.

---

netstat Command

Purpose

Shows network status.

Syntax

To Display Active Sockets for Each Protocol or Routing Table Information

/bin/netstat [**-n**] [{ **-A** **-a** } | { **-r** **-C** **-i** **-I** *Interface* }] [**-f** *AddressFamily*] [**-p** *Protocol*] [*Interval*] [*System*]

To Display the Contents of a Network Data Structure

/bin/netstat [**-m** | **-s** | **-ss** | **-u** | **-v**] [**-f** *AddressFamily*] [**-p** *Protocol*] [*Interval*] [*System*]

To Display the Packet Counts Throughout the Communications Subsystem

/bin/netstat -D

To Display the Network Buffer Cache Statistics

/bin/netstat -c

To Display the Data Link Provider Interface Statistics

/bin/netstat -P

To Clear the Associated Statistics

/bin/netstat [**-Zc** | **-Zi** | **-Zm** | **-Zs**]

Description

The **netstat** command symbolically displays the contents of various network-related data structures for active connections. The *Interval* parameter, specified in seconds, continuously displays information regarding packet traffic on the configured network interfaces. The *Interval* parameter takes no flags. The *System* parameter specifies the memory used by the current kernel. Unless you are looking at a dump file, the *System* parameter should be **/unix**.

Flags

-A Shows the address of any protocol control blocks associated with the sockets. This flag acts with the default display and is used for debugging purposes.

-a Shows the state of all sockets. Without this flag, sockets used by server processes are not shown.

-c Shows the statistics of the Network Buffer Cache.

The Network Buffer Cache is a list of network buffers that contain data objects that can be transmitted to networks. The Network Buffer Cache grows dynamically as data objects are added to or removed from it. The Network Buffer Cache is used by some network kernel interfaces for performance enhancement on the network I/O. The **netstat -c** command prints the following statistic:

```
Network Buffer Cache Statistics:
Current total cache buffer size: 0
Maximum total cache buffer size: 0
Current total cache data size: 0
Maximum total cache data size: 0
Current number of cache: 0
Maximum number of cache: 0
Number of cache with data: 0
Number of searches in cache: 0
Number of cache hit: 0
Number of cache miss: 0
Number of cache newly added: 0
Number of cache updated: 0
Number of cache removed: 0
Number of successful cache accesses: 0
Number of unsuccessful cache accesses: 0
Number of cache validation: 0
Current total cache data size in private segments: 0
Maximum total cache data size in private segments: 0
Current total number of private segments: 0
Maximum total number of private segments: 0
Current number of free private segments: 0
Current total NBC_NAMED_FILE entries: 0
Maximum total NBC_NAMED_FILE entries: 0
```

Note: The **-c** flag is only valid on AIX 4.3.2 and above.

-C Shows the routing tables, including the user-configured and current costs of each route. The user-configured cost is set using the **-hopcount** flag of the **route** command. The current cost may be different than the user-configured cost if Dead Gateway Detection has changed the cost of the route.

-D Shows the number of packets received, transmitted, and dropped in the communications subsystem.

Note: In the statistics output, a `N/A` displayed in a field value indicates the count is not applicable. For the NFS/RPC statistics, the number of incoming packets that pass through RPC are the same packets that pass through NFS, so these numbers are not summed in the `NFS/RPC Total` field, thus the `N/A`. NFS has no outgoing packet or outgoing packet drop counters specific to NFS and RPC. Therefore, individual counts have a field value of `N/A`, and the cumulative count is stored in the `NFS/RPC Total` field.

-f *AddressFamily* Limits reports of statistics or address control blocks to those items specified by the *AddressFamily* variable. The following address families are recognized:

inet Indicates the AF_INET address family.

inet6 Indicates the AF_INET6 address family.

ns Indicates the AF_NS address family.

unix Indicates the AF_UNIX address family.

-i Shows the state of all configured interfaces. See ″Interface Display.″
Note: The collision count for Ethernet interfaces is not supported.

-I *Interface* Shows the state of the configured interface specified by the *Interface* variable.

-m Shows statistics recorded by the memory management routines.

-n Shows network addresses as numbers. When this flag is not specified, the **netstat** command interprets addresses where possible and displays them symbolically. This flag can be used with any of the display formats.

-p *Protocol* Shows statistics about the value specified for the *Protocol* variable, which is either a well-known name for a protocol or an alias for it. Some protocol names and aliases are listed in the **/etc/protocols** file. A null response means that there are no numbers to report. The program report of the value specified for the *Protocol* variable is unknown if there is no statistics routine for it.

-P Shows the statistics of the Data Link Provider Interface (DLPI). The **netstat -P** command prints the following statistic:

```
DLPI statistics:
Number of received packets = 0
Number of transmitted packets = 0
Number of received bytes = 0
Number of transmitted bytes = 0
Number of incoming pkts discard = 0
Number of outgoing pkts discard = 0
Number of times no buffers = 0
Number of successful binds = 0
Number of unknown message types = 0
Status of phys level promisc = 0
Status of sap level promisc = 0
Status of multi level promisc = 0
Number of enab_multi addresses = 0
```

If DLPI is not loaded, it displays:

```
can't find symbol: dl_stats
```

Note: The **-P** flag is only valid on AIX 4.3.2 and above.

-r Shows the routing tables. When used with the **-s** flag, the **-r** flag shows routing statistics. See ″Routing Table Display.″

-s Shows statistics for each protocol.

-ss Displays all the non-zero protocol statistics and provides a concise display.

-u Displays information about domain sockets.

-v Shows statistics for CDLI-based communications adapters. This flag causes the **netstat** command to run the statistics commands for the **entstat**, **tokstat**, and **fddistat** commands. No flags are issued to these device driver commands. See the specific device driver statistics command to obtain descriptions of the statistical output.

-Zc Clear network buffer cache statistics.

-Zi Clear interface statistics.

-Zm Clear network memory allocator statistics.

-Zs Clear protocol statistics. To clear statistics for a specific protocol, use -p <protocol>. For example, to clear TCP statistics, type `netstat -Zs -p tcp`.

Default Display

The default display for active sockets shows the following items:

- Local and remote addresses
- Send and receive queue sizes (in bytes)
- Protocol
- Internal state of the protocol

Internet address formats are of the form `host.port` or `network.port` if a socket's address specifies a network but no specific host address. The host address is displayed symbolically if the address can be resolved to a symbolic host name, while network addresses are displayed symbolically according to the **/etc/networks** file.

NS addresses are 12-byte quantities, consisting of a 4-byte network number, a 6-byte host number and a 2-byte port number, all stored in network standard format. For VAX architecture, these are word and byte reversed; for the Sun systems, they are not reversed.

If a symbolic name for a host is not known or if the **-n** flag is used, the address is printed numerically, according to the address family. Unspecified addresses and ports appear as an * (asterisk).

Interface Display (netstat -i)

The interface display format provides a table of cumulative statistics for the following items:

- Errors
- Collisions

  Note: The collision count for Ethernet interfaces is not supported.
- Packets transferred

The interface display also provides the interface name, number, and address as well as the maximum transmission units (MTUs).

Routing Table Display (netstat -r)

The routing table display indicates the available routes and their statuses. Each route consists of a destination host or network and a gateway to use in forwarding packets.

A route is given in the format *A.B.C.D*/*XX*, which presents two pieces of information. *A.B.C.D* indicates the destination address and *XX* indicates the netmask associated with the route. The netmask is represented by the number of bits set. For example, the route `9.3.252.192/26` has a netmask of `255.255.255.192`, which has 26 bits set.

The routing table contains the following ten fields:

`Flags`

The **flags** field of the routing table shows the state of the route:

A	An Active Dead Gateway Detection is enabled on the route. This field only applies to AIX 5.1 or later.
U	Up.
H	The route is to a host rather than to a network.
G	The route is to a gateway.
D	The route was created dynamically by a redirect.
M	The route has been modified by a redirect.
L	The link-level address is present in the route entry.
c	Access to this route creates a cloned route. This field only applies to AIX 4.2.1 or later.
W	The route is a cloned route. This field only applies to AIX 4.2.1 or later.
1	Protocol specific routing flag #1.
2	Protocol specific routing flag #2.
3	Protocol specific routing flag #3.
b	The route represents a broadcast address.
e	Has a binding cache entry.
I	The route represents a local address.
m	The route represents a multicast address.
P	Pinned route.
R	Host or net unreachable.
S	Manually added.
u	Route usable.
s	The Group Routing stopsearch option is enabled on the route.

Direct routes are created for each interface attached to the local host.

`Gateway` The **gateway** field for these entries shows the address of the outgoing interface.

`Refs` Gives the current number of active uses for the route. Connection-oriented protocols hold on to a single route for the duration of a connection, while connectionless protocols obtain a route while sending to the same destination.

`Use` Provides a count of the number of packets sent using that route.

`PMTU` Gives the Path Maximum Transfer Unit (PMTU). This field only applies to AIX 4.2.1 or later.

`Interface` Indicates the network interfaces utilized for the route.

`Exp` Displays the time (in minutes) remaining before the route expires. This field only applies to AIX 4.2.1 or later.

`Groups` Provides a list of group IDs associated with that route. This field only applies to AIX 4.2.1 or later.

`Netmasks` Lists the netmasks applied on the system.

`Route Tree for Protocol Family`

Specifies the active address families for existing routes. Supported values for this field are:

1 Specifies the UNIX address family.

2 Specifies the Internet address family (for example, TCP and UDP).

6 Specifies the Xerox Network System (XNS) address family.

For more information on other address families, refer to the **/usr/include/sys/socket.h** file.

When a value is specified for the *Interval* parameter, the **netstat** command displays a running count of statistics related to network interfaces. This display contains two columns: a column for the primary interface (the first interface found during autoconfiguration) and a column summarizing information for all interfaces.

The primary interface may be replaced with another interface by using the **-I** flag. The first line of each screen of information contains a summary of statistics accumulated since the system was last restarted. The subsequent lines of output show values accumulated over intervals of the specified length.

Inet Examples

1. To display routing table information for an Internet interface, type:

   ```
   netstat -r -f inet
   ```

   This produces the following output:

   ```
   Routing tables
   Destination   Gateway      Flags Refs Use  PMTU If  Exp Groups Netmasks:
   (root node)
   (0)0 ffff f000 0
   (0)0 ffff f000 0
   (0)0 8123 262f 0 0 0 0 0
   (root node)

   Route Tree for Protocol Family 2:
   (root node)
   default        129.35.38.47   UG    0  564   -   tr0   -
   loopback       127.0.0.1      UH    1  202   -   lo0   -
   129.35.32      129.35.41.172  U     4   30   -   tr0   -   +staff
   129.35.32.117  129.35.41.172  UGHW  0   13  1492 tr0   30
   192.100.61     192.100.61.11  U     1  195   -   en0   -
   (root node)

   Route Tree for Protocol Family 6:
   (root node)
   (root node)
   ```

   The `-r -f inet` flags indicate a request for routing table information for all configured Internet interfaces. The network interfaces are listed in the `Interface` column; `en` designates a Standard Ethernet interface, while `tr` specifies a Token-Ring interface. Gateway addresses are in dotted decimal format.

2. To display interface information for an Internet interface, type:

   ```
   netstat -i -f inet
   ```

   This produces the following output if you are using AIX 4.2:

   ```
   Name Mtu   Network     Address   Ipkts  Ierrs  Opkts  Oerrs  Coll
   lo0  1536  <Link>                  4      0      4      0     0
   lo0  1536  127         loopback    4      0      4      0     0
   en0  1500  <Link>                 96      0     67      0     0
   ```

```
en0  1500  192.100.61  nullarbor    96      0     67      0     0
tr0  1500  <Link>               44802      0  11134      0     0
tr0  1500  129.35.32   stnullarb 44802      0  11134      0     0
```

This produces the following output if you are using AIX 4.3:

```
Name Mtu    Network     Address           Ipkts  Ierrs  Opkts  Oerrs  Coll
lo0  16896  Link#1                        5161      0    5193      0     0
lo0  16896  127         localhost         5161      0    5193      0     0
lo0  16896  ::1                           5161      0    5193      0     0
en1  1500   Link#2      8.0.38.22.8.34    221240    0  100284      0     0
en1  1500   129.183.64  infoserv.frec.bul 221240    0  100284      0     0
```

The `-i -f inet` flags indicate a request for the status of all configured Internet interfaces. The network interfaces are listed in the `Name` column; `lo` designates a loopback interface, `en` designates a Standard Ethernet interface, while `tr` specifies a Token-Ring interface.

3. To display statistics for each protocol, type:

```
netstat -s -f inet
```

This produces the following output:

```
ip:
:
  44485 total packets received
  0 bad header checksums
  0 with size smaller than minimum
  0 with data size < data length
  0 with header length < data size
  0 with data length < header length
  0 with bad options
  0 with incorrect version number
  0 fragments received
  0 fragments dropped (dup or out of space)
  0 fragments dropped after timeout
  0 packets reassembled ok
  44485 packets for this host
  0 packets for unknown/unsupported protocol
  0 packets forwarded
  0 packets not forwardable
  0 redirects sent
  1506 packets sent from this host
  0 packets sent with fabricated ip header
  0 output packets dropped due to no bufs, etc.
  0 output packets discarded due to no route
  0 output datagrams fragmented
  0 fragments created
  0 datagrams that can't be fragmented
  0 IP Multicast packets dropped due to no receiver
  0 successful path MTU discovery cycles
  0 path MTU rediscovery cycles attempted
  0 path MTU discovery no-response estimates
  0 path MTU discovery response timeouts
  0 path MTU discovery decreases detected
  0 path MTU discovery packets sent
  0 path MTU discovery memory allocation failures
  0 ipintrq overflows

icmp:
  0 calls to icmp_error
  0 errors not generated 'cuz old message was icmp
  Output histogram:
    echo reply: 6
  0 messages with bad code fields
  0 messages < minimum length
  0 bad checksums
  0 messages with bad length
```

```
  Input histogram:
    echo: 19
  6 message responses generated

igmp:defect
  0 messages received
  0 messages received with too few bytes
  0 messages received with bad checksum
  0 membership queries received
  0 membership queries received with invalid field(s)
  0 membership reports received
  0 membership reports received with invalid field(s)
  0 membership reports received for groups to which we belong
  0 membership reports sent

tcp:
  1393 packets sent
    857 data packets (135315 bytes)
    0 data packets (0 bytes) retransmitted
    367 URG only packets
    0 URG only packets
    0 window probe packets
    0 window update packets
    170 control packets
  1580 packets received
    790 acks (for 135491 bytes)
    60 duplicate acks
    0 acks for unsent data
    638 packets (2064 bytes) received in-sequence
    0 completely duplicate packets (0 bytes)
    0 packets with some dup. data (0 bytes duped)
    117 out-of-order packets (0 bytes)
    0 packets (0 bytes) of data after window
    0 window probes
    60 window update packets
    0 packets received after close
    0 discarded for bad checksums
    0 discarded for bad header offset fields
  0 connection request
  58 connection requests
  61 connection accepts
  118 connections established (including accepts)
  121 connections closed (including 0 drops)
  0 embryonic connections dropped
  845 segments updated rtt (of 847 attempts)
  0 resends due to path MTU discovery
  0 path MTU discovery terminations due to retransmits
  0 retransmit timeouts
    0 connections dropped by rexmit timeout
  0 persist timeouts
  0 keepalive timeouts
    0 keepalive probes sent
    0 connections dropped by keepalive

udp:
  42886 datagrams received
:
  0 incomplete headers
  0 bad data length fields
  0 bad checksums
  0 dropped due to no socket
  42860 broadcast/multicast datagrams dropped due to no

socket
  0 socket buffer overflows
  26 delivered
  106 datagrams output
```

`ip` specifies the Internet Protocol; `icmp` specifies the Information Control Message Protocol; `tcp` specifies the Transmission Control Protocol; `udp` specifies the User Datagram Protocol.

4. To display device driver statistics, type:

   ```
   netstat -v
   ```

   The `netstat -v` command displays the statistics for each CDLI-based device driver that is up. To see sample output for this command, see the **tokstat** command, the **entstat** command, or the **fddistat** command.

5. To display information regarding an interface for which multicast is enabled, and to see group membership, type:

   ```
   netstat -a -I interface
   ```

   For example, if an 802.3 interface was specified, the following output will be produced:

   ```
   Name  Mtu  Network Address      Ipkts  Ierrs  Opkts  Oerrs  Coll
   et0   1492 <Link>                 0      0      2      0     0
   et0   1492 9.4.37  hun-eth          0      0      2      0     0
                      224.0.0.1
                      02:60:8c:0a:02:e7
                      01:00:5e:00:00:01
   ```

   If instead of **-I** *interface* the flag **-i** is given, then all configured interfaces will be listed. The network interfaces are listed in the Name column; **lo** designates a loopback interface, **et** designates an IEEE 802.3 interface, **tr** designates a Token-Ring interface, while **fi** specifies an FDDI interface.

   The address column has the following meaning. A symbolic name for each interface is shown. Below this symbolic name, the group addresses of any multicast groups that have been joined on that interface are shown. Group address 224.0.0.1 is the special *all-hosts-group* to which all multicast interfaces belong. The MAC address of the interface (in colon notation) follows the group addresses, plus a list of any other MAC level addresses that are enabled on behalf of IP Multicast for the particular interface.

6. To display the packet counts in the communication subsystem, type:

   ```
   netstat -D
   ```

   The following output will be produced:

   ```
   Source                    Ipkts     Opkts     Idrops      Odrops
   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   tok_dev0                   720       542       0          0
   ent_dev0                   114         4       0          0
                     - - - - - - - - - - - - - - - - - - - - - - - - -
   Devices Total              834       546       0          0
   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   tok_dd0                    720       542       0          0
   ent_dd0                    114         4       0          0
                     - - - - - - - - - - - - - - - - - - - - - - - - -
   Drivers Total              834       546       0          0
   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   tok_dmx0                   720       N/A       0          N/A
   ent_dmx0                   114       N/A       0          N/A
                     - - - - - - - - - - - - - - - - - - - - - - - - -
   Demuxer Total              834       N/A       0          N/A
   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   IP                         773       767       0          0
   TCP                        536       399       0          0
   UDP                        229        93       0          0
                     - - - - - - - - - - - - - - - - - - - - - - - - -
   Protocols Total           1538      1259       0          0
   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   lo_if0                     69        69       0          0
   en_if0                     22         8       0          0
   ```

```
tr_if0                    704       543       0          1
                  - - - - - - - - - - - - - - - - - - - - - - -
Net IF Total              795       620       0          1
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
NFS/RPC Client            519       N/A       0          N/A
NFS/RPC Server              0       N/A       0          N/A
NFS Client                519       N/A       0          N/A
NFS Server                 0       N/A       0          N/A
                  - - - - - - - - - - - - - - - - - - - - - - -
NFS/RPC Total             N/A       519       0          0
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(Note:  N/A -> Not Applicable)
```

Xerox Network System (XNS) Examples

1. To display network information for an XNS interface, type:

   ```
   netstat -i -f ns
   ```

   This produces the following output:

   ```
   Name  Mtu  Network Address      Ipkts  Ierrs  Opkts  Oerrs  Coll
   en1   1500 ns:6EH  2608C2EA9F7H  281   0       3055   0      0
   et1   1492 ns:78H  2608C2EA9F7H  44    0       3043   0      0
   nsip0 1536 ns:1H   2608C2EA9F7H  0     0       0      0      0
   ```

   The `-i -f ns` flags indicate a request for the status of all configured XNS interfaces. The network interfaces are listed in the `Name` column; `en` designates a Standard Ethernet interface, while `et` specifies an IEEE 802.3 Ethernet interface. The `ns:` in the `Network` column designates the XNS family address. All network and address numbers are in hexadecimal with the letter `H` appended to the end of the number.

   The `nsip0` is the Internet encapsulated XNS packet. The Internet destination address used for encapsulation is specified in the `ipdst` field in the **ifconfig** command.

2. To display routing table information for an XNS interface, type:

   ```
   netstat -r -f ns
   ```

   This produces the following output:

   ```
   Routing tables
   Destination      Gateway           Flags  Refcnt  Use  Interface
   Route Tree for Protocol Family 6:
   (root node)
   1H.2608C2EA394H  1H.2608C2EA9F7H   UH     1        0   nsip0
   18H.*            78H.2608C2EA9F7H  UG     0        0   et1
   6EH.*            6EH.2608C2EA9F7H  U      1        0   en1
   78H.*            78H.2608C2EA9F7H  U      1        0   et1
   (root node)
   ```

   The `-r -f ns` flags indicate a request for routing table information for all configured XNS interfaces. The network interfaces are listed in the `Interface` column; `en` designates a Standard Ethernet interface, while `et` specifies an IEEE 802.3 Ethernet interface. The `nsip0` in the `Interface` column designates an XNS to Internet encapsulation interface. All `Destination` and `Gateway` address numbers are in hexadecimal with the letter `H` appended to the end of the number. The * (asterisk) in the `Destination` column indicates the network is not a point-to-point network.

Related Information

The **atmstat** command, **entstat** command, **fddistat** command, **iostat** command, **tokstat** command, **trpt** command, **vmstat** command.

The **hosts** file format, **networks** file format, **protocols** file format, **services** file format.

Monitoring and Tuning Communications I/O in *AIX 5L Version 5.2 Performance Management Guide*.

Gateways, Naming, TCP/IP Addressing, TCP/IP Network Interfaces, TCP/IP Protocols, and TCP/IP Routing in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

Xerox Network Systems (XNS) Overview for Programming in *AIX 5L Version 5.2 Communications Programming Concepts*.

---

newaliases Command

Purpose

Builds a new copy of the alias database from the mail aliases file.

Syntax

newaliases

Description

The **newaliases** command builds a new copy of the alias database from the **/etc/aliases** file. It must be run each time this file is changed in order for the changes to take effect. Running this command is equivalent to running the **sendmail** command with the **-bi** flag.

Exit Status

0	Exits successfully.
>0	An error occurred.

Files

/usr/sbin/newaliases	Contains the **newaliases** command.
/etc/mailaliases	Contains source for the mail aliases file command.
/etc/aliasesDB directory	Contains the binary files created by the **newaliases** command.

Related Information

The **sendmail** command.

Managing Mail Aliases and How to Build the Alias Database in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

---

newform Command

Purpose

Changes the format of a text file.

Syntax

newform [**-s**] [[**-f**] [**-a** [*Number*]] [**-b** [*Number*]] [**-c** [*Character*]] [**-e** [*Number*]] [**-i** [*TabSpec*]] [**-l** [*Number*]] [**-o** [*TabSpec*]] [**-p** [*Number*]] [*File* ...]] ...

Description

The **newform** command takes lines from the files specified by the *File* parameter (standard input by default) and writes the formatted lines to standard output. Lines are reformatted in accordance with the command-line flags in effect.

Except for the **-s** flag, you can enter command-line flags in any order, repeated, and mixed with the *File* parameter. However, the system processes command-line flags in the order you specify. For example, the **-c** flag modifies the behavior of the **-a** and **-p** flags, so specify the **-c** flag before the **-p** or **-a** flag for which it is intended. The **-l** (lowercase L) flag modifies the behavior of the **-a**, **-b**, **-e**, and **-p** flags, so specify the **-l** flag before the flags for which it is intended. For example, flag sequences like **-e**15 **-l**60 yield results that are different from **-l**60 **-e**15. Flags are applied to all files specified on the command line.

An exit value of 0 indicates normal execution; an exit value of 1 indicates an error.

Notes:

1. The **newform** command normally only keeps track of physical characters; however, for the **-i** and **-o** flags, the **newform** command keeps track of backspaces to line up tabs in the appropriate logical columns.
2. The **newform** command does not prompt you if the system reads a *TabSpec* variable value from standard input (by use of the **-i-** or **-o-** flag).
3. If you specify the **-f** flag, and the last **-o** flag you specified was **-o-** preceded by either an **-o-** or an **-i-**, the tab-specification format line is incorrect.
4. If the values specified for the **-p**, **-l**, **-e**, **-a**, or **-b** flag are not valid decimal numbers greater than 1, the specified value is ignored and default action is taken.

Flags

-a [*Number*]
: Adds the specified number of characters to the end of the line when the line length is less than the effective line length. If no number is specified, the **-a** flag defaults to 0 and adds the number of characters necessary to obtain the effective line length. See also the **-c** [*Character*] and **-p** [*Number*] flags.

-b [*Number*]
: Truncates the specified number of characters from the beginning of the line if the line length is greater than the effective line length. If the line also contains fewer characters than specified by the *Number* parameter, the entire line is deleted and a blank line is displayed in its place. See also the **-l** [*Number*] flag. If you specify the **-b** flag with no *Number* variable, the default action truncates the number of characters necessary to obtain the effective line length.

  This flag can be used to delete the sequence numbers from a COBOL program, as follows:

  ```
  newform -l1-b7 file-name
  ```

  The **-l1** flag must be used to set the effective line length shorter than any existing line in the file so that the **-b** flag is activated.

-c [*Character*]
: Changes the prefix/add character to that specified by the *Character* variable. Default character is a space and is available when specified before the **-a** and **-p** flags.

-e [*Number*]
: Truncates the specified number of characters from the end of the line. Otherwise, the flag is the same as the **-b** [*Number*] flag.

-f
: Writes the tab-specification format line to standard output before any other lines are written. The displayed tab-specification format line corresponds to the format specified by the final **-o** flag. If no **-o** flag is specified, the line displayed contains the default specification of -8.

-i [*TabSpec*] Replaces all tabs in the input with the number of spaces specified by the *TabSpec* variable.

This variable recognizes all tab specification forms described in the **tabs** command.

If you specify a **-** (minus sign) for the value of the *TabSpec* variable, the **newform** command assumes that the tab specification can be found in the first line read from standard input. The default *TabSpec* value is -8. A *TabSpec* value of -0 expects no tabs. If any are found, they are treated as having a value of -1.

-l [*Number*] Sets the effective line length to the specified number of characters. If no *Number* variable is specified, the **-l** flag defaults to 72. The default line length without the **-l** flag is 80 characters. Note that tabs and backspaces are considered to be one character (use the **-i** flag to expand tabs to spaces). You must specify the **-l** flag before the **-b** and **-e** flags.

-o [*TabSpec*] Replaces spaces in the input with a tab in the output, according to the tab specifications given. The default *TabSpec* value is -8. A *TabSpec* value of -0 means that no spaces are converted to tabs on output.

-p [*Number*] Appends the specified number of characters to the beginning of a line when the line length is less than the effective line length. The default action is to append the number of characters that are necessary to obtain the effective line length. See also the **-c** flag.

-s Removes leading characters on each line up to the first tab and places up to 8 of the removed characters at the end of the line. If more than 8 characters (not counting the first tab) are removed, the 8th character is replaced by an ***** (asterisk) and any characters to the right of it are discarded. The first tab is always discarded.

The characters removed are saved internally until all other specified flags are applied to that line. The characters are then added to the end of the processed line.

Note: The values for the **-a**, **-b**, **-e**, **-l** (lowercase L), and **-p** flags cannot be larger than **LINE_MAX** or 2048 bytes.

Examples

To convert from a file with:

- Leading digits
- One or more tabs
- Text on each line

to a file:

- Beginning with the text, all tabs after the first expanded to spaces
- Padded with spaces out to column 72 (or truncated to column 72)
- Leading digits placed starting at column 73

type the following:

```
newform -s -i -l -a -e filename
```

The **newform** command displays the following error message and stops if the **-s** flag is used on a file without a tab on each line.

```
newform: 0653-457 The file is not in a format supported by the -s flag.
```

Related Information

The **tabs** command, **csplit** command.

newgrp Command

Purpose

Changes a user's real group identification.

Syntax

newgrp [**-**] [**-l**] [*Group*]

Description

The **newgrp** command changes a user's real group identification. When you run the command, the system places you in a new shell and changes the name of your real group to the group specified with the *Group* parameter. By default, the **newgrp** command changes your real group to the group specified in the **/etc/passwd** file.

> **Note:** The **newgrp** command does not take input from standard input and cannot be run from within a script.

The **newgrp** command recognizes only group names, not group ID numbers. Your changes only last for the current session. You can only change your real group name to a group you are already a member of. If you are a root user, you can change your real group to any group regardless of whether you are a member of it or not.

> **Note:** When you run the **newgrp** command, the system always replaces your shell with a new one. The command replaces your shell regardless of whether the command is successful or not. For this reason, the command does not return error codes.

Flags

- Changes the environment to the login environment of the new group.
-l Indicates the same value as the **-** flag.

Security

Access Control: This command should be installed as a program in the trusted computing base (TCB). The command should be owned by the root user with the **setuid** (SUID) bit set.

Exit Status

If the **newgrp** command succeeds in creating a new shell execution environment, regardless if the group identification was changed successfully, the exit status will be that of the current shell. Otherwise, the following exit value is returned:

>0 An error occurred.

Examples

1. To change the real group ID of the current shell session to `admin`, enter:
   ```
   newgrp admin
   ```
2. To change the real group ID back to your original login group, enter:
   ```
   newgrp
   ```

Files

/etc/group
Indicates the group file; contains group IDs.

/etc/passwd
Indicates the password file; contains user IDs.

Related Information

The **login** command, **setgroups** command.

---

newkey Command

Purpose

Creates a new key in the **/etc/publickey** file.

Syntax

/usr/sbin/newkey [**-h** *HostName*] [**-u** *UserName*]

Description

The **newkey** command creates a new key in the **/etc/publickey** file. This command is normally run by the network administrator on the Network Information Services (NIS) master machine to establish public keys for users and root users on the network. These keys are needed for using secure Remote Procedure Call (RPC) protocol or secure Network File System (NFS).

The **newkey** command prompts for the login password of the user specified by the *UserName* parameter. Then, the command creates a new key pair in the **/etc/publickey** file and updates the **publickey** database. The key pair consists of the user's public key and secret key and is encrypted with the login password of the given user.

Use of this program is not required. Users may create their own keys using the **chkey** command.

You can use the Network application in Web-based System Manager (wsm) to change network characteristics. You could also use the System Management Interface Tool (SMIT) **smit newkey** fast path to run this command.

Flags

-h *HostName* Creates a new public key for the root user at the machine specified by the *HostName* parameter. Prompts for the root password of this parameter.

-u *UserName* Creates a new public key for a user specified by the *UserName* parameter. Prompts for the NIS password of this parameter.

Examples

1. To create a new public key for a user, enter:

   ```
   newkey -u john
   ```

   In this example, the **newkey** command creates a new public key for the user named `john`.
2. To create a new public key for the root user on host `zeus`, enter:

   ```
   newkey -h zeus
   ```

In this example, the **newkey** command creates a new public key for the root user on the host named zeus.

Files

/etc/publickey	Stores encrypted keys for users.

Related Information

The **chkey** command, **keylogin** command.

The **keyserv** daemon.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.2 Web-based System Manager Administration Guide*.

System Management Interface Tool in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices*.

Network File System (NFS) Overview for System Management in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

Exporting a File System Using Secure NFS, Mounting a File System Using Secure NFS in *AIX 5L Version 5.2 Security Guide*.

Network Information Service (NIS) in *AIX 5L Version 5.2 Network Information Services (NIS and NIS+) Guide*.

NIS Reference.

---

news Command

Purpose

Writes system news items to standard output.

Syntax

news [**-a** | **-n** | **-s** | *Item ...*]

Description

The **news** command writes system news items to standard output. This command keeps you informed of news concerning the system. Each news item is contained in a separate file in the **/var/news** directory. Most users run the **news** command followed by the **-n** flag each time they log in by including it in their **$HOME/.profile** file or in the system's **/etc/profile** file. Any user having write permission to this directory can create a news item. It is not necessary to have read permission to create a news item.

If you run the **news** command without any flags, it displays every current file in the **/var/news** file, showing the most recent first. This command, used with the **-a** flag, displays all news items. If you specify the **-n** flag, only the names of the unread news items are displayed. Using the **-s** flag displays the number of unread news items. You can also use the *Item* parameter to specify the files that you want displayed.

Each file is preceded by an appropriate header. To avoid reporting old news, the **news** command stores a currency time. The **news** command considers your currency time to be the date the **$HOME/.news_time**

file was last modified. Each time you read the news, the modification time of this file changes to that of the reading. Only news item files posted after this time are considered current.

Pressing the Interrupt (Ctrl-C) key sequence during the display of a news item stops the display of that item and starts the next. Pressing the Ctrl-C key sequence again ends the **news** command.

Note: News items can contain multibyte characters.

Flags

-a Displays all news items, regardless of the currency time. The currency time does not change.
-n Reports the names of current news items without displaying their contents. The currency time does not change.
-s Reports the number of current news items without displaying their names or contents. The currency time does not change.

Examples

1. To display the items that have been posted since you last read the news, enter:
   ```
   news
   ```
2. To display all the news items, enter:
   ```
   news -a | pg
   ```
   All of the news items display a page at a time (`| pg`), regardless of whether you have read them yet.
3. To list the names of the news items that you have not read yet, enter:
   ```
   news -n
   ```
   Each name is a file in the **/var/news** directory.
4. To display specific news items, enter:
   ```
   news newusers services
   ```
   This command sequence displays news about `newusers` and `services`, which are names listed by the **news -n** command.
5. To display the number of news items that you have not yet read, enter:
   ```
   news -s
   ```
6. To post news for everyone to read, enter:
   ```
   cp schedule /var/news
   ```
   This copies the `schedule` file into the system **/var/news** directory to create the **/var/news/schedule** file. To do this, you must have write permission to the **/var/news** directory.

Files

/usr/bin/news	Contains the **news** command.
/etc/profile	Contains the system profile.
/var/news	Contains system news item files.
$HOME/.news_time	Indicates the date the **news** command was last invoked.

Related Information

The **pg** command.

The **/etc/security/environ** file, **profile** file.

---

next Command

Purpose

Shows the next message.

Syntax

next [**+***Folder*] [**-header** | **-noheader**] [**-showproc** *CommandString* | **-noshowproc**]

Description

The **next** command displays the number the system will assign to the next message filed in a Message Handler (MH) folder. The **next** command is equivalent to the **show** command with the **next** value specified as the message.

The **next** command links to the **show** program and passes any switches on to the **showproc** program. If you link to the **next** value and call that link something other than **next**, your link will function like the **show** command, rather than like the **next** command.

The **show** command passes flags it does not recognize to the program performing the listing. The **next** command provides a number of flags for the listing program.

Flags

+*Folder*	Specifies the folder that contains the message you want to show.
-header	Displays a one-line description of the message being shown. The description includes the folder name and message number. This is the default.
-help	Lists the command syntax, available switches (toggles), and version information. **Note:** For MH, the name of this flag must be fully spelled out.
-noheader	Prevents display of a one-line description of each message being shown.
-noshowproc	Uses the **/usr/bin/cat** file to perform the listing. This is the default.
-showproc *CommandString*	Uses the specified command string to perform the listing.

Examples

1. To see the next message in the current folder, enter:

   ```
   next
   ```

   The system responds with a message similar to the following:

   ```
   (Message schedule: 10)
   ```

   The text of the message is also displayed. In this example, message `10` in the current folder `schedule` is the next message.
2. To see the next message in the `project` folder, enter:

   ```
   next  +project
   ```

The system responds with the text of the message and a header similar to the following:

```
(Message project: 5)
```

Files

$HOME/.mh_profile	Specifies a user's MH profile.
/usr/bin/next	Contains the **next** command.

Related Information

The **prev** command, **show** command.

The **.mh_alias** file format, **.mh_profile** file format.

Mail Overview in *AIX 5L Version 5.2 System User's Guide: Communications and Networks*.

---

nfsd Daemon

Purpose

Services client requests for file system operations.

Syntax

/usr/sbin/nfsd [**-a** | **-p** { *tcp* | *udp* }] [**-c** *max_connections*] [**-w** *max_write_size*] [**-r** *max_read_size*] *nservers*

Description

The **nfsd** daemon runs on a server and handles client requests for file system operations.

Each daemon handles one request at a time. Assign the maximum number of threads based on the load you expect the server to handle.

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

```
startsrc -s nfsd
stopsrc -s nfsd
```

To change the number of daemons started with the SRC commands, use the **chnfs** command. To change the parameters of an SRC controlled daemon, use the **chssys** command.

Flags

-a	Specifies UDP and TCP transport will be serviced. This flag only applies to AIX 4.2.1 or later.
-c *max_connections*	Specifies the maximum number of TCP connections allowed at the NFS server. This flag only applies to AIX 4.2.1 or later.
nservers	Specifies the maximum number of concurrent requests that the NFS server can handle. This concurrency is achieved by the number of *nservers* thread created in the kernel as needed. This flag only applies to AIX 4.2.1 or later.
-p *tcp* or **-p** *udp*	Transports both UDP and TCP to the NFS clients (default). You can only specify UDP or TCP. For example, if **-p** *tcp* is used, the NFS server only accepts NFS client requests using the TCP protocol. This flag only applies to AIX 4.2.1 or later.

-r *max_read_size* Specifies for NFS Version 3, the maximum size allowed for file read requests. The default and maximum allowed is 32K. This flag only applies to AIX 4.2.1 or later.

-w *max_write_size* Specifies for NFS Version 3, the maximum size allowed for file write requests. The default and maximum allowed is 32K. This flag only applies to AIX 4.2.1 or later.

Parameter

Parameter that can be changed:

NumberOfNfsds Specifies the number of daemons to start. This parameter does not apply to AIX 4.2.1 or later.

Examples

1. To start **nfsd** daemons using an **src** command, enter:

   ```
   startsrc -s nfsd
   ```

   In this example, the `startsrc -s nfsd` entry starts the number of daemons specified in the script.
2. To change the number of daemons running on your system, enter:

   ```
   chssys -s nfsd -a 6
   ```

   In this example, the `chssys` command changes the number of `nfsd` daemons running on your system to 6.

Related Information

The **chnfs** command, **chssys** command.

The **biod** daemon, **mountd** daemon.

Network File System (NFS) Overview for System Management in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices*.

System Resource Controller Overview in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*.

List of NFS Commands.

---

nfso Command

Purpose

Manages Network File System (NFS) tuning parameters.

Syntax

nfso [**-p** | **-r**] [**-c**] { **-o** *Tunable*[**=***Newvalue*] }

nfso [**-p** | **-r**] { **-d** *Tunable* }

nfso [**-p** | **-r**] **-D**

nfso [**-p** | **-r**] **-a** [**-c**]

nfso -?

nfso -h *Tunable*

nfso -l *Hostname*

nfso [**-c**]

nfso -L [*Tunable*]

Note: Multiple flags **-o**, **-d**, and **-L** are allowed.

Description

Use the **nfso** command to configure Network File System tuning parameters. The **nfso** command sets or displays current or next boot values for Network File System tuning parameters. This command can also make permanent changes or defer changes until the next reboot. Whether the command sets or displays a parameter is determined by the accompanying flag. The **-o** flag performs both actions. It can either display the value of a parameter or set a new value for a parameter.

Attention: Be careful when you use this command. If used incorrectly, the **nfso** command can make your system inoperable.

Flags

-a Displays the current, reboot (when used in conjunction with **-r**) or permanent (when used in conjunction with **-p**) value for all tunable parameters, one per line in pairs *Tunable* **=** *Value*. For the permanent options, a value is only displayed for a parameter if its reboot and current values are equal. Otherwise `NONE` displays as the value.

-c Changes the output format of the **nfso** command to colon-delineated format.

-d *Tunable* Sets the *Tunable* variable back to its default value. If a *Tunable* needs to be changed that is, . it is currently not set to its default value) and is of type Bosboot or Reboot, or if it is of type Incremental and has been changed from its default value, and **-r** is not used in combination, it will not be changed but a warning displays instead.

-D Sets all *Tunable*s variable back to their default value. If *Tunable* needing to be changed are of type Bosboot or Reboot, or are of type Incremental and have been changed from their default value, and **-r** is not used in combination, they will not be changed but warnings display instead.

-h *Tunable* Displays help about the *Tunable* parameter.

-l *HostName* Allows a system administrator to release NFS file locks on an NFS server. The *HostName* variable specifies the host name of the NFS client that has file locks held at the NFS server. The **nfso -l** command makes a remote procedure call to the NFS server's **rpc.lockd** network lock manager to request the release of the file locks held by the *HostName* NFS client.

If there is an NFS client that has file locks held at the NFS server and this client has been disconnected from the network and cannot be recovered, the **nfso -l** command can be used to release those locks so that other NFS clients can obtain similar file locks for their purposes.
Note: The **nfso** command can be used to release locks on the local NFS server only.

-o *Tunable*[**=***NewValue*]
Displays the value or sets *Tunable* to *NewValue*. If a tunable needs to be changed (the specified value is different than current value), and is of type Bosboot or Reboot, or if it is of type Incremental and its current value is bigger than the specified value, and **-r** is not used in combination, it will not be changed but a warning displays instead.

When **-r** is used in combination without a new value, the nextboot value for the *Tunable* displays. When **-p** is used in combination without a *NewValue*, a value displays only if the current and next boot values for the *Tunable* are the same. Otherwise `NONE` displays as the value.

-p
Makes changes apply to both current and reboot values, when used in combination with **-o**, **-d** or **-D**, that is, it turns on the updating of the **/etc/tunables/nextboot** file in addition to the updating of the current value. These combinations cannot be used on Reboot and Bosboot type parameters because their current value cannot be changed.

When used with **-a** or **-o** without specifying a new value, values are displayed only if the current and next boot values for a parameter are the same. Otherwise `NONE` displays as the value.

-r
Makes changes apply to reboot values when used in combination with **-o**, **-d** or **-D**, that is, it turns on the updating of the **/etc/tunables/nextboot** file. If any parameter of type Bosboot is changed, the user is prompted to run bosboot.

When used with **-a** or **-o** without specifying a new value, next boot values for tunables display instead of current values.

-L *Tunable*
Lists the characteristics of one or all *Tunable*, one per line, using the following format:

```
Name   Current  Default  Reboot   Minimum  Maximum  Unit  Type Dependencies
                             value    value    value    value    value
-------------------------------------------------------------------------

param1 5        2        4        1        10       MB/s     I
```

Parameters can be of the following types:
- D for *Dynamic* if tunable can be changed at any time.
- S for *Static* if tunable can never be changed.
- R for *Reboot* if tunable can only be changed during reboot sequence.
- B for *Bosboot* if tunable can only be changed by running bosboot and rebooting machine.
- M for *Mount* if tunable is only effective for future filesystems or directory mountings.
- I for *Incremental* if tunable can only be incremented, except at boot time.

Any change (with **-o**, **-d**, or **-D**) to a parameter of type Mount results in a message displaying to warn the user that the change is only effective for future mountings.

Any attempt to change (with **-o**, **-d**, or **-D**) a parameter of type Bosboot or Reboot without **-r**, results in an error message.

Any attempt to change (with **-o**, **-d**, or **-D** but without **-r**) the current value of a parameter of type Incremental with a new value smaller than the current value, results in an error message.

Note: The current set of parameters managed by the **nfso** command include *Dynamic*, *Mount*, and *Incremental* types.

-?
Displays the **nfso** usage statement.

Compatibility Mode

When running in pre 5.2 compatibility mode (controlled by the **pre520tune** attribute of **sys**0, see Introduction to AIX 5L Version 5.2 tunable parameter setting in the Performance Management Guide), reboot values for parameters, except those of type Bosboot, are not really meaningful because in this mode they are not applied at boot time.

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

This mode is automatically turned ON when a machine is MIGRATED to AIX 5L Version 5.2. For complete installations, it is turned OFF and the reboot values for parameters are set by applying the content of the **/etc/tunables/nextboot** file during the reboot sequence. Only in that mode are the **-r** and **-p** flags fully functional. See AIX 5L Version 5.2 kernel tuning in the Performance Tools Guide and Reference for details about the new 5.2 mode.

Tunable Parameters

nfs_allow_all_signals

Purpose:

Specifies that the NFS server adhere to signal handling requirements for blocked locks for the UNIX 95/98 test suites.

Values:

Default: 0

Range: 0 or 1

Type: Dynamic

Diagnosis:

N/A

Tuning:

A value of 1 turns nfs_allow_all_signals on, and a value of 0 turns it off.

nfs_device_specific_bufs (AIX 4.2.1 and later)

Purpose:

This option allows the NFS server to use memory allocations from network devices if the network device supports such a feature.

Values:

Default: 1

Range: 0 or 1

Type: Dynamic

Diagnosis:

N/A

Tuning:

Use of these special memory allocations by the NFS server can positively affect the overall performance of the NFS server. The default of 1 means the NFS server is allowed to use the special network device memory allocations. If the value of 0 is used, the NFS server uses the traditional memory allocations for its processing of NFS client requests. These are buffers managed by a network interface that result in improved performance (over regular mbufs) because no setup for DMA is required on these. Two adapters that support this include the Micro Channel ATM adapter and the SP2 switch adapter.

nfs_dynamic_retrans

Purpose:

Specifies whether the NFS client should use a dynamic retransmission algorithm to decide when to resend NFS requests to the server.

Values:

Default: 1

Range: 0 or 1

Type: Dynamic

Diagnosis:

N/A

Tuning:

If this function is turned on, the timeo parameter is only used in the first retransmission. With this parameter set to 1, the NFS client attempts to adjust its timeout behavior based on past NFS server response. This allows for a floating timeout value along with adjusting the transfer sizes used. All of this is done based on an accumulative history of the NFS server's response time. In most cases, this parameter does not need to be adjusted. There are some instances where the straightforward timeout behavior is desired for the NFS client. In these cases, the value should be set to 0 before mounting file systems.

Refer to:

Tuning to Avoid Retransmits

nfs_gather_threshold

Purpose:

Sets the minimum size of write requests for which write gathering is done.

Values:

Default: 4096

Useful Range: 512 to 8193

Type: Dynamic

Diagnosis:

N/A

Tuning:

One of the following two situations exists:

1. Delays are observed in responding to RPC requests, particularly those where the client is exclusively doing non-sequential writes or the files being written are being written with file locks held on the client.
2. Clients are writing with write sizes < 4096 and write-gather is not working. If write-gather is to be disabled, change the nfs_gather_threshold to a value greater than the largest possible write. For AIX Version 4 running NFS Version 2, this value is 8192. Changing the value to 8193 disables write gather. Use this for the situation described above in scenario (1). If write gather is being bypassed due to a small write size, say 1024, change the write gather parameter to gather smaller writes; for example, set to 1024.

nfs_iopace_pages (AIX 4.1)

Purpose:

Specifies the number of NFS file pages that are scheduled to be written back to the server through the VMM at one time. This I/O scheduling control occurs on close of a file and when the system invokes the *syncd* daemon.

Values:

Default: 0 (32 before AIX 4.2.1)

Range: 0 to 65536

Type: Dynamic

Diagnosis:

N/A

Tuning:

When an application writes a large file to an NFS mounted filesystem, that file data is written to the NFS server when the file is closed. In some cases, the resource it takes to write that file to the server may prevent other NFS file I/O from occurring. This parameter limits the number of 4k pages written to the server to the value of nfs_iopace_pages. The NFS client will schedule nfs_iopace_pages for writing to the server and then will wait for these to complete before scheduling the next batch of pages. The default value will usually be sufficient for most environments. Decreased the values if there are large amounts of contention for NFS client resources. If there is low contention, the value can be increased. In AIX 4.2.1 and later, if nfs_iopace_pages=0, then the number of pages written by the **syncd** daemon at one time is as follows:

```
MAX ((filesize/8)-1, 32)
```

nfs_max_connections

Purpose:

Specifies the maximum number of TCP connections allowed into the server.

Values:

Default: 0 (indicates no limit)

Range: 0 10 10000

Type: Dynamic

Diagnosis:

N/A

Tuning:

Limits number of connections into the server in order to reduce load.

Refer to:

Tuning Other Layers to Improve NFS Performance

nfs_max_threads (AIX 4.2.1 and later)

Purpose:

Specifies the maximum number of NFS server threads that are created to service incoming NFS requests.

Values:

Default: 3891

Range: 5 to 5000

Type: Dynamic

Diagnosis:

With AIX 4.2.1, the NFS server is multithreaded. The NFS server threads are created as demand increases for the NFS server. When the NFS server threads become idle, they will exit. This allows the server to adapt to the needs of the NFS clients. The nfs_max_threads parameter is the maximum number of threads that can be created.

Tuning:

In general, it does not detract from overall system performance to have the maximum set to something very large because the NFS server creates threads as needed. However, this assumes that NFS-serving is the primary machine purpose. If the desire is to share the system with other activities, then the maximum number of threads may need to be set low. The maximum number can also be specified as a parameter to the nfsd daemon.

Refer to:

How Many biod and nfsd Daemons Are Needed

nfs_repeat_messages (AIX Version 4)

Purpose:

Checks for duplicate NFS messages. This option is used to avoid displaying duplicate NFS messages.

Values:

Default: 0 (no)

Range: 0 or 1

Type: Dynamic

Diagnosis:

N/A

Tuning:

Tuning this parameter does not affect performance.

nfs_rfc1323 (AIX 4.3)

Purpose:

Enables very large TCP window size negotiation (greater than 65535 bytes) to occur between systems.

Values:

Default: 0

Range: 0 or 1

Type: Dynamic

Diagnosis:

N/A

Tuning:

If using the TCP transport between NFS client and server, and both systems support it, this allows the systems to negotiate a TCP window size in a way that allows more data to be *in-flight* between the client and server. This increases the throughput potential between client and server. Unlike the rfc1323 option of the no command, this only affects NFS and not other applications in the system. Value of 0 means this is disabled, and value of 1 means it is enabled. If the no command parameter rfc1323 is already set, this NFS option does not need to be set.

nfs_server_base_priority

Purpose:

Sets the base priority of **nfsd** daemons.

Values:

Default: 65

Range: 31 to 125

Type: Dynamic

Diagnosis:

N/A

Tuning:

By default, the **nfsd** daemons run with a floating process priority. Therefore, as they increase their cumulative CPU time, their priority changes. This parameter can be used to set a static parameter for the **nfsd** daemons. The value of 0 represents the floating priority (default). Other values within the acceptable range are used to set the priority of the **nfsd** daemon when an NFS request is received at the server. This option can be used if the NFS server is overloading the system (lowering or making the nfsd daemon less favored). It can also be used if you want the nfsd daemons be one of the most favored processes on the server. Use caution when setting the parameter because it can render the system almost unusable by other processes. This situation can occur if the NFS server is very busy and essentially locks out other processes from having run time on the server.

nfs_server_clread (AIX 4.2.1 and later)

Purpose:

This option allows the NFS server to be very aggressive about the reading of a file. The NFS server can only respond to the specific NFS-read request from the NFS client. However, the NFS server can read data in the file which exists immediately after the current read request. This is normally referred to as read-ahead. The NFS server does read-ahead by default.

Values:

Default: 1

Range: 0 or 1

Type: Dynamic

Diagnosis:

May be useful in cases where server memory is low and too much disk-to-memory activity is occurring.

Tuning:

With the nfs_server_clread option enabled, the NFS server becomes very aggressive about doing read-ahead for the NFS client. If value is 1, then aggressive read-ahead is done; If value is 0, normal system default read-ahead methods are used. Normal system read-ahead is controlled by VMM. In AIX 4.2.1, the more aggressive top-half JFS read-ahead was introduced. This mechanism is less susceptible to read-ahead breaking down due to out-of-order requests (which are typical in the NFS server case). When the mechanism is activated, it will read an entire cluster (128 KB, the LVM logical track group size).

nfs_setattr_error (AIX 4.2.1 and later)

Purpose:

When enabled, NFS server ignores setattr requests that are not valid.

Values:

Default: 0 (disabled)

Range: 0 or 1

Type: Dynamic

Diagnosis:

N/A

Tuning:

This option is provided for certain PC applications. Tuning this parameter does not affect performance.

nfs_socketsize

Purpose:

Sets the queue size of the NFS server UDP socket.

Values:

Default: 60000

Practical Range: 60000 to 204800

Type: Dynamic

Diagnosis:

N/A

Tuning:

Increase the size of the nfs_socketsize variable when netstat reports packets dropped due to full socket buffers for UDP, and increasing the number of nfsd daemons has not helped.

Refer to:

Increasing NFS Socket Buffer Size

nfs_tcp_duplicate_cache_size (AIX 4.2.1 and later)

Purpose:

Specifies the number of entries to store in the NFS server's duplicate cache for the TCP network transport.

Values:

Default: 5000

Range: 1000 to 100000

Type: Incremental

Diagnosis:

N/A

Tuning:

The duplicate cache size cannot be decreased. Increase the duplicate cache size for servers that have a high throughput capability. The duplicate cache is used to allow the server to correctly respond to NFS client retransmissions. If the server flushes this cache before the client is able to retransmit, then the server may respond incorrectly. Therefore, if the server can process 1000 operations before a client retransmits, the duplicate cache size must be increased.

Calculate the number of NFS operations that are being received per second at the NFS server and multiply this by 4. The result is a duplicate cache size that should be sufficient to allow correct response from the NFS server. The operations that are affected by the duplicate cache are the following: **setattr**(), **write**(), **create**(), **remove**(), **rename**(), **link**(), **symlink**(), **mkdir**(), **rmdir**().

nfs_tcp_socketsize (AIX 4.2.1 and later)

Purpose:

Sets the queue size of the NFS server TCP socket. The queue size is specified in number of bytes. The TCP socket is used for receiving the NFS client requests and can be adjusted so that the NFS server is less likely to drop packets under a heavy load. The value of the nfs_tcp_socketsize option must be less than the sb_max option, which can be manipulated by the no command.

Values:

Default: 60000

Practical Range: 60000 to sb_max

Type: Dynamic

Diagnosis:

Packets dropped when examining the output of the command **netstat -s -p tcp**.

Tuning:

This option reserves, but does not allocate, memory for use by the send and receive socket buffers of the socket. Do not set the nfs_tcp_socketsize value to less than 60,000. Large or busy servers should have larger values until TCP NFS traffic shows no packets dropped from the output of the **netstat -s -p tcp** command.

Refer to:

Tuning Other Layers to Improve NFS Performance

nfs_udp_duplicate_cache_size (AIX 4.2.1 and later)

Purpose:

Specifies the number of entries to store in the NFS server's duplicate cache for the UDP network transport.

Values:

Default: 5000

Range: 1000 to 100000

Type: Incremental

Diagnosis:

N/A

Tuning:

The duplicate cache size cannot be decreased. Increase the duplicate cache size for servers that have a high throughput capability. The duplicate cache is used to allow the server to correctly respond to NFS client retransmissions. If the server flushes this cache before the client is able to retransmit, then the server may respond incorrectly. Therefore, if the server can process 1000 operations before a client retransmits, the duplicate cache size must be increased.

Calculate the number of NFS operations that are being received per second at the NFS server and multiply this by 4. The result is a duplicate cache size that should be sufficient to allow correct response from the NFS server. The operations that are affected by the duplicate cache are the following: **setattr**(), **write**(), **create**(), **remove**(), **rename**(), **link**(), **symlink**(), **mkdir**(), **rmdir**().

nfs_use_reserved_ports (AIX 4.2.1 and later)

Purpose:

Specifies using nonreserved IP port number.

Values:

Default: 0

Range: 0 or 1

Type: Dynamic

Diagnosis:

N/A

Tuning:

Value of 0 use a nonreserved IP port number when the NFS client communicates with the NFS server.

portcheck

Purpose:

Checks whether an NFS request originated from a privileged port.

Values:

Default: 0

Range: 0 or 1

Type: Dynamic

Diagnosis:

N/A

Tuning:

Value of 0 disables the port-checking that is done by the NFS server. A value of 1 directs the NFS server to do port checking on the incoming NFS requests. This is a configuration decision with minimal performance consequences.

udpchecksum

Purpose:

Turns on or off the generation of checksums on NFS UDP packets.

Values:

Default: 1

Range: 0 or 1

Type: Dynamic

Diagnosis:

N/A

Tuning:

Make sure this value is set to on in any network where packet corruption might occur. Slight performance gains can be realized by turning it off, but at the expense of increased chance of data corruption.

Refer to:

nfs_tcp_socketsize

nfs_v2_pdts

Purpose:

Sets the number of tables for memory pools used by the biods for NFS Version 2 mounts.

Values:

Default: 1

Range: 1 to 8

Type: Mount

Diagnosis:

N/A

Tuning:

N/A

nfs_v3_pdts

Purpose:

Sets the number of tables for memory pools used by the biods for NFS Version 3 mounts.

Values:

Default: 1

Range: 1 to 8

Type: Mount

Diagnosis:

N/A

Tuning:

N/A

nfs_v2_vm_bufs

Purpose:

Sets the number of initial free memory buffers used for each NFS version 2 Paging Device Table (pdt) created after the first table. The very first pdt has a set value of 256, 512, 640 or 1000, depending on system memory. This initial value is also the default value of each newly created pdt.
Note: The initial set value for the first pdt table will never change.

Values:

Default: 1000

Range: 512 to 5000

Type: Dynamic

Diagnosis:

N/A

Tuning:

N/A

nfs_v3_vm_bufs

Purpose:

Sets the number of initial free memory buffers used for each NFS version 3 Paging Device Table(pdt) created after the first table. The very first pdt has a set value of 256, 512, 640 or 1000, depending on system memory. This initial value is also the default value of each newly created pdt.
Note: The initial set value for the first pdt table will never change.

Values:

Default: 1000

Range: 512 to 5000

Type: Dynamic

Diagnosis:

N/A

Tuning:

N/A

Examples

1. To set the **portcheck** tunable parameter to a value of zero, type:

   ```
   nfso -o portcheck=0
   ```

2. To set the **udpchecksum** tunable parameter to its default value of 1 at the next reboot, type:

   ```
   nfso -r -d udpchecksum
   ```

3. To print, in colon-delimited format, a list of all tunable parameters and their current values, type:

   ```
   nfso -a -c
   ```

4. To list the current and reboot value, range, unit, type and dependencies of all tunables parameters managed by the **nfso** command, type:

   ```
   nfso -L
   ```

5. To display help information on nfs_tcp_duplicate_cache_size, type:

   ```
   nfso -h nfs_tcp_duplicate_cache_size
   ```

6. To permanently turn off nfs_dynamic_retrans, type:

   ```
   nfso -p -o nfs_dynamic_retrans=0
   ```

7. To list the reboot values for all Network File System tuning parameters, type:

   ```
   no -r -a
   ```

Related Information

The **netstat** command, **no** command, **vmo** command, **ioo** command, **schedo** command, **tunsave** command, **tunrestore** command, **tuncheck** command, and **tundefault** command.

Network File System (NFS) Overview for System Management in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

TCP/IP Overview for System Management in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

Monitoring and Tuning NFS Use in *AIX 5L Version 5.2 Performance Management Guide*.

List of NFS Commands.

---

nfsstat Command

Purpose

Displays statistical information about the Network File System (NFS) and Remote Procedure Call (RPC) calls.

Syntax

/usr/sbin/nfsstat [**-c**] [**-s**] [**-n**] [**-r**] [**-z**] [**-m**]

Description

The **nfsstat** command displays statistical information about the NFS and Remote Procedure Call (RPC) interfaces to the kernel. You can also use this command to reinitialize this information. If no flags are given, the default is the **nfsstat -csnr** command. With this option, the command displays everything, but reinitializes nothing.

RPC Server Information

The server RPC display includes the following fields:

`calls` Total number of RPC calls received

badcalls	Total number of calls rejected by the RPC layer
nullrecv	Number of times an RPC call was not available when it was thought to be received
badlen	Number of RPC calls with a length shorter than a minimum-sized RPC call
xdrcall	Number of RPC calls whose header could not be XDR decoded
dupchecks	Number of RPC calls that looked up in the duplicate request cache
dupreqs	Number of duplicate RPC calls found

RPC Client Information

calls	Total number of RPC calls made
badcalls	Total number of calls rejected by the RPC layer
badxid	Number of times a reply from a server was received that did not correspond to any outstanding call
timeouts	Number of times a call timed out while waiting for a reply from the server
newcreds	Number of times authentication information had to be refreshed
badverfs	The number of times the call failed due to a bad verifier in the response.
timers	The number of times the calculated time-out value was greater than or equal to the minimum specified timed-out value for a call.
cantconn	The number of times the call failed due to a failure to make a connection to the server.
nomem	The number of times the calls failed due to a failure to allocate memory.
interrupts	The number of times the call was interrupted by a signal before completing.
retrans	The number of times a call had to be retransmitted due to a time-out while waiting for a reply from the server. This is applicable only to RPC over connection-less transports
dupchecks	The number of RPC calls that looked up in the duplicate request cache.
dupreqs	The number of duplicate RPC calls found.

NFS Server Information

The NFS server displays the number of NFS calls received (`calls`) and rejected (`badcalls`), as well as the counts and percentages for the various kinds of calls made.

NFS Client Information

The NFS client information displayed shows the number of calls sent and rejected, as well as the number of times a CLIENT handle was received (`clgets`), the number of times the client handle had no unused entries (`clatoomany`), and a count of the various kinds of calls and their respective percentages.

-m Information

The **-m** flag displays information about **mount** flags set by **moun**t options, **mount** flags internal to the system, and other **mount** information. See the **mount** command for more information.

The following **mount** options are set by **mount** flags:

auth	Provides one of the following values: **none** No authentication. **unix** UNIX style authentication (UID, GID). **des** des style authentication (encrypted timestamps).
hard	Hard mount.
soft	Soft mount.
intr	Interrupts allowed on hard mount.
nointr	No interrupts allowed on hard mount.
noac	Client is not catching attributes.
rsize	Read buffer size in bytes.
wsize	Write buffer size in bytes.
retrans	NFS retransmissions.
nocto	No close-to-open consistency.

`llock`	Local locking being used (no lock manager.
`grpid`	Group ID inheritance.
`vers`	NFS version.
`proto`	Protocol.

The following **mount** options are internal to the system:

`printed`	Not responding message printed.
`down`	Server is down.
`dynamic`	Dynamic transfer size adjustment.
`link`	Server supports links.
`symlink`	Server supports symbolic links.
`readdir`	Use **readdir** instead of **readdirplus**.

Flags

-c Displays client information. Only the client side NFS and RPC information is printed. Allows the user to limit the report to client data only. The **nfsstat** command provides information about the number of RPC and NFS calls sent and rejected by the client. To print client NFS or RPC information only, combine this flag with the **-n** or **-r** option.

-m Displays statistics for each NFS file system mounted along with the server name and address, mount flags, current read and write sizes, retransmission count, and the timers used for dynamic retransmission. This flag only applies to AIX 4.2.1 or later.

-n Displays NFS information . Prints NFS information for both the client and server. To print only the NFS client or server information, combine this flag with the **-c** and **-s** options.

-r Displays RPC information.

-s Displays server information.

-z Reinitializes statistics. This flag is for use by the root user only and can be combined with any of the above flags to zero particular sets of statistics after printing them.

Examples

1. To display information about the number of RPC and NFS calls sent and rejected by the client, enter:
   ```
   nfsstat -c
   ```
2. To display and print the client NFS call-related information, enter:
   ```
   nfsstat -cn
   ```
3. To display statistics for each NFS mounted file system in AIX 4.2.1 or later, enter:
   ```
   nfsstat -m
   ```
4. To display and print RPC call-related information for the client and server, enter:
   ```
   nfsstat -r
   ```
5. To display information about the number of RPC and NFS calls received and rejected by the server, enter:
   ```
   nfsstat -s
   ```
6. To reset all call-related information to zero on the client and server, enter:
   ```
   nfsstat -z
   ```

   Note: You must have root user authority to use the **-z** flag.

Related Information

Network File System (NFS) Overview for System Management in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

List of NFS Commands in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

Monitoring and Tuning NFS Use in *AIX 5L Version 5.2 Performance Management Guide*.

---

nice Command

Purpose

Runs a command at a lower or higher priority .

Syntax

nice [**-** *Increment*| **-n** *Increment*] *Command* [*Argument ...*]

Description

The **nice** command lets you run a command at a priority lower than the command's normal priority. The *Command* parameter is the name of any executable file on the system. If you do not specify an *Increment* value the **nice** command defaults to an increment of 10. You must have root user authority to run a command at a higher priority. The priority of a process is often called its nice value.

The nice value can range from 0 to 39, with 39 being the lowest priority. For example, if a command normally runs at a priority of 20, specifying an increment of 5 runs the command at a lower priority, 25, and the command runs slower. The **nice** command does not return an error message if you attempt to increase a command's priority without the appropriate authority. Instead, the command's priority is not changed, and the system starts the command as it normally would.

The nice value is used by the system to calculate the current priority of a running process. Use the **ps** command with the **-l** flag to view a command's nice value. The nice value appears under the **NI** heading in the **ps** command output.

> **Note:** The **csh** command contains a built-in command named **nice**. The **/usr/bin/nice** command and the **csh** command's **nice** command do not necessarily work the same way. For information on the **csh** command's **nice** command, see the **csh** command.

Flags

-*Increment*	Increments a command's priority up or down. You can specify a positive or negative number. Positive increment values reduce priority. Negative increment values increase priority. Only users with root authority can specify a negative increment. If you specify an increment value that would cause the nice value to exceed the range of 0 to 39, the nice value is set to the value of the limit that was exceeded. This flag is equivalent to the **-n** *Increment* flag.
-n *Increment*	This flag is equivalent to the **-***Increment* flag.

Exit Status

If the command specified by the *Command* parameter is started, the exit status of the **nice** command is the exit status of the command specified by the *Command* parameter. Otherwise, the **nice** command exits with one of the following values:

1-125	An error occurred in the **nice** command.
126	The command specified by the *Command* parameter was found but could not be invoked.
127	The command specified by the *Command* parameter could not be found.

Examples

1. To specify a very low priority, enter:

   ```
   nice -n 15 cc -c *.c &
   ```

   This example runs the **cc** command in the background at a lower priority than the default priority set by the **nice** command.

2. To specify a very high priority, enter:

   ```
   nice --10 wall <<end
   System shutdown in 2 minutes!
   end
   ```

   This example runs the **wall** command at a higher priority than all user processes, which slows down everything else running on the system. The `<<end` and `end` portions of the example define a *here document*, which uses the text entered before the end line as standard input for the command.

   Note: If you do not have root user authority when you run this command, the **wall** command runs at the normal priority.

3. To run a command at low priority, enter:

   ```
   nice cc -c *.c
   ```

   This example runs the **cc** command at low priority.

   Note: This does not run the command in the background. The workstation is not available for doing other things.

4. To run a low-priority command in the background, enter:

   ```
   nice cc -c *.c &
   ```

   This example runs the **cc** command at low priority in the background. The workstation is free to run other commands while the **cc** command is running. Refer to the ″Shells Overview″ in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices* for more information on background (asynchronous) processing.

Files

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

Related Information

The **csh** command, **nohup** command, **renice** command.

The **nice** subroutine.

The Processes Overview in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices* discusses what processes are and input and output redirection.

Shells Overview in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices* describes what shells are, the different types, and how they affect the way commands are interpreted.

Controlling Contention for the CPU in *AIX 5L Version 5.2 Performance Management Guide*.

nim Command

Purpose

Performs operations on Network Installation Management (NIM) objects.

Syntax

nim { **-o** *Operation}* [**-F**] [**-t** *Type*] [**-a** *Attribute*=*Value* . . .] *{ObjectName}*

Description

The **nim** command performs an operation on a NIM object. The type of operation performed is dependent on the type of object specified by the *ObjectName* parameter. Possible operations include initializing environments and managing resources. You can use the **lsnim** command to display the list of supported operations.

Flags

-a *Attribute* **=** *Value* . . .	Assigns the specified value to the specified attribute. Use the **lsnim -q** *Operation* **-t** *Type* command to get a list of valid attributes for a specific operation.
-F	Overrides some safety checks.

-o *Operation*

Specifies an operation to perform on a NIM object. The possible operations are:

allocate
Allocates a resource for use.

bos_inst
Performs a BOS installation.

change
Changes an object's attributes.

check Checks the status of a NIM object.

cust Performs software customization.

deallocate
Deallocates a resource.

define Defines an object.

diag Enables a machine to boot a diagnostic image.

dkls_init
Initializes a diskless machine's environment.

dtls_init
Initializes a dataless machine's environment.

fix_query
Lists the fix information for a given APAR or keyword.

lppchk Verifies installed filesets on NIM machines and SPOTs

lslpp Lists licensed program information about an object.

maint Performs software maintenance.

maint_boot
Enables a machine to boot in maintenance mode.

remove
Removes an object.

reset Resets an object's NIM state.

showlog
Displays a NIM client's installation, boot or customization log, or a SPOT's installation log from the NIM master.

showres
Displays the contents of a NIM resource.

sync_roots
Synchronizes root directories for diskless and dataless clients for a specific Shared Product Object Tree (SPOT).

unconfig
Unconfigures the NIM master fileset.

Use the **lsnim -POt** *Type* command to get a list of the valid operations for a specific type.

AIX 4.3 or later Operations:

alt_disk_install
Performs an alternate disk installation.

-t *Type*

Specifies the type of the NIM object for define operations. The possible types are:

resource types:

bosinst_data
Config file used during base system installation.

dump Parent directory for client dump files.

exclude_files
Contains files to be excluded from a **mksysb** image. This resource applies only to AIX 4.2 or later.

fix_bundle
Fix (keyword) input file for the **cust** or **fix_query** operation.

home Parent directory for client **/home** directories.

image_data
Config file used during base system installation.

installp_bundle
Installp bundle file.

lpp_source
Source device for optional product images.

mksysb
mksysb image.

paging Parent directory for client paging files.

root Parent directory for client **/** (root) directories.

resolv_conf
Name-server configuration file.

script Executable file that is run on a client.

shared_home
/home directory shared by clients.

spot Shared Product Object Tree (SPOT) - equivalent to **/usr** file system.

tmp Parent directory for client **/tmp** directories.

machine types:

diskless
All file systems and resources remote.

dataless
Local paging,dump; remote **/**,**/usr**; others remote or local.

standalone
Local file systems and resources.

master Machine that controls the NIM environment.

-t *Type* (Continued)

network types:

tok Token-Ring network.

ent Ethernet network.

fddi FDDI network.

atm ATM network. (AIX 4.3 or later.)

generic
Other TCP/IP networks.

group types:

mac_group
Group of machines.

res_group
Group of resources.

Security

Access Control: You must have root authority to run the **nim** command.

Examples

The following examples are grouped by operation.

define

1. To define an rspc uniprocessor dataless machine for AIX 5.1 and earlier on the token-ring network called `net1` and call it `altoid`, type:

   ```
   nim -o define -t dataless -a if1="net1 fred 10005aa88500" \
   -a ring_speed=16 -a platform=rspc -a netboot_kernel=up \
   -a comments="Dataless client altoid"
   ```

   The `comments` attribute is optional and may contain any user-entered notes.

   Note: The if1 attribute is required.

2. To define a resource that is a directory containing installable images that is located on server `altoid` and has a path name of `/usr/sys/inst.images`, and name that resource `images`, type:

   ```
   nim -o define -t lpp_source -a server=altoid \
   -a location=/usr/sys/inst.images images
   ```

3. To create a new SPOT resource named `myspot` on the NIM master in the `/export/exec directory`, using `an` **lpp_source** named `images`, type:

   ```
   nim -o define -t spot -a server=master -a location=/export/exec \
   -a source=images myspot
   ```

4. To define a network object named `BLDG905`, with a subnetmask of `255.255.240.0` and an address of `129.35.129.0`, type:

   ```
   nim -o define -t tok -a snm=255.255.240.0 \
   -a net_addr=129.35.129.0 BLDG905
   ```

5. To define an **lpp_source**, `lppsrc1`, that will be located on the master from a tape selecting a specific set of software products that are on the tape, `bos.INed` and `bos.adt`, type:

   ```
   nim -o define -t lpp_source -a location=/images2/lppsrc1 \
   -a source=/dev/rmt0 -a server=master -a packages="bos.INed \
   bos.adt" lppsrc1
   ```

6. To define a standalone machine that is a Symmetrical Multi-Processor model for AIX 5.1 and earlier that has a BOOTP-enabled IPL ROM, its hostname is `jupiter` and it will also be known as `jupiter` in the NIM environment, using a token-ring network called `net1` and a ring speed of `16`, type:

```
nim -o define -t standalone -a if1="net1 jupiter 0" \
-a ring_speed=16 -a platform=rs6ksmp jupiter
```

7. To define a **mksysb** resource, mksysb1, from an existing mksysb image located in /resources/mksysb.image on the master, type:

```
nim -o define -t mksysb -a server=master \
-a location=/resources/mksysb.image mksysb1
```

8. To define a NIM network named ATMnet with a subnet mask of 255.255.240 and an address of 129.35.101.0 to represent an ATM network, use the generic network type as follows:

```
nim -o define -t generic -a snm=255.255.240.0 \
-a net.addr=129.35.101.0 ATMnet
```

9. To define a PowerPC PCI bus-based, symmetric multiprocessor computer for AIX 5.1 and earlier whose hostname is bluefish as a standalone machine on a token ring network called net1 and have the machine be known to the NIM environment as bluefish, type:

```
nim -o define -t standalone -a platform=rspcsmp \
-a if1="net1 bluefish 0" -a ring_speed=16 bluefish
```

10. To define a machine group named DisklsMacs1 with members that are NIM diskless machines named diskls1, diskls2, and diskls3, type:

```
nim -o define -t mac_group -a add_member=diskls1 \
-a add_member=diskls2 -a add_member=diskls3 DisklsMacs1
```

11. To define a resource group named DisklsRes1 with resources spot1, root1, dump1, paging1, home1, tmp1, type:

```
nim -o define -t res_group -a spot=spot1 -a root=root1 \
-a dump=dump1 -a paging=paging1 -a home=home1 -a tmp=tmp1 \
DisklsRes1
```

12. To display the space required to define a **mksysb** resource, mksysb2, and create a mksysb image of the client, client1, during the resource definition where the image will be located in /resources/mksysb.image on the master, type:

    Note: This action only shows the space required for the operation, mksysb or resource creation does NOT take place.

```
nim -o define -t mksysb -a server=master \
-a location=/resources/mksysb.image -a source=client1 \
-a mk_image=yes -a size_preview=yes mksysb2
```

13. To define a **mksysb** resource, mksysb2, and create the mksysb image of the client, client1, during the resource definition where the image will be located in /resources/mksysb.image on the master, type:

```
nim -o define -t mksysb -a server=master \
-a location=/resources/mksysb.image -a source=client1 \
-a mk_image=yes mksysb2
```

14. To define a **mksysb** resource, mksysb2, and create a mksysb image of the client, client1, during the resource definition where the mksysb flags used to create the image are **-em**, and the image will be located in /resources/mksysb.image on the master, type:

```
nim -o define -t mksysb -a server=master \
-a location=/resources/mksysb.image -a source=client1 \
-a mk_image=yes -a mksysb_flags=em mksysb2
```

15. To define an **exclude_files** resource, exclude_file1, located in /resources/exclude_file1 on the master, type:

```
nim -o define -t exclude_files -a server=master \
-a location=/resources/exclude_file1 exclude_file1
```

16. A machine called redfish, hostname redfish_t.lab.austin.ibm.com, has its primary interface attached to a token-ring network with ring speed of 16 Megabits. To define redfish as a standalone machine in the NIM environment and instructing NIM to locate the name of the network that the machine's primary interface is attached, type:

```
nim -o define -t standalone  -a if1="find_net \
redfish_t.lab.austin.ibm.com 0" -a ring_speed1=16 redfish
```

17. A machine called `bluefish`, hostname is `bluefish_e.lab.austin.ibm.com`, has its primary interface attached to an ethernet network with **cable_type** of **bnc**. To define `bluefish` as a diskless machine in the NIM environment and instructing NIM to locate the name of the network that the machine's primary interface is attached, and if not found, have NIM define a NIM network with the name `ent_net`, subnetmask of `255.255.255.128` and default route using the gateway with hostname `lab_gate`, type:

    ```
    nim -o define -t diskless -a if1="find_net \
    bluefish_e.lab.austin.ibm.com 0" -a net_definition="ent \
    255.255.255.128 lab_gate 0 ent_net" -a cable_type=bnc bluefish
    ```

    Note: Specify 0 in place of the master gateway in the **net_definition** attribute if a default route for the master already exists, otherwise you must specify the master gateway.

remove

To remove a resource named `dump_files`, type:

```
nim -o remove dump_files
```

change

1. Machines on the `BLDG905` network use the `gateway905` gateway to reach the `OZ` network. Machines on the `OZ` network use the `gatewayOZ` gateway to reach the `BLDG905` network. To add a route between two networks named `BLDG905` and `OZ`, type:

    ```
    nim -o change -a routing1="OZ gateway905 gatewayOZ" BLDG905
    ```

2. The adapter identified by the host name `sailfish2.austin.ibm.com` is attached to a token ring network. To define a secondary interface for this adapter on the NIM master and instructing NIM to locate the NIM network representing the attached ethernet network and, if not found, have NIM define a NIM network with subnetmask `255.255.255.128`, type:

    ```
    nim -o change -a if2="find_net sailfish2.austin.ibm.com 0" \
    -a net_definition="tok 255.255.255.128" -a ring_speed2=16 master
    ```

    Note: A default name is generated for the network, and no routing information is specified for the new network.

3. To define default routes for the networks `net1` and `net2` that use default gateways `gw1` and `gw2` respectively, type the following two commands:

    ```
    nim -o change -a routing1="default gw1" net1
    nim -o change -a routing1="default gw2" net2
    ```

4. To designate the resources defined by the resource group `res_grp1` as the set of resources always allocated by default during any operation in which these resources are applicable, type:

    ```
    nim -o change -a default_res=res_grp1 master
    ```

check

1. To have NIM check on the usability of a SPOT named `myspot`, type:

    ```
    nim -o check myspot
    ```

2. To check the status of an **lpp_source** named `images`, type:

    ```
    nim -o check images
    ```

allocate

1. To allocate resources to a diskless workstation with the name of `syzygy` and SPOT attribute value of `spot1`, type:

    ```
    nim -o allocate -a spot=spot1 syzygy
    ```

2. To perform a base system installation on the machine named `krakatoa`, resources must first be allocated by typing:

    ```
    nim -o allocate -a spot=myspot -a lpp_source=images krakatoa
    ```

Then the NIM environment can be initialized to support the install by performing the **bos_inst** operation, type:

```
nim -o bos_inst krakatoa
```

3. To install the software product, `adt`, into a standalone machine, `stand1`, given that the installable option, `adt`, resides in the **lpp_source**, `images`, type:

   ```
   nim -o allocate -a lpp_source=images stand1
   ```

   Then type:

   ```
   nim -o cust -a filesets="adt" stand1
   ```

4. To install software products into a standalone machine, `stand1`, such that the image for the installable option, `adt`, resides in the **lpp_source**, `images`, and the **installp_bundle**, `bundle1`, contains the name of the installable option, type:

   ```
   nim -o allocate -a lpp_source=images \
   -a installp_bundle=bundle1 stand1
   ```

   Then type:

   ```
   nim -o cust stand1
   ```

5. To automatically configure a machine with name resolution services after a BOS installation, create the file `/exports/resolv.conf`, with contents similar to the following:

   ```
   nameserver      129.35.143.253
   nameserver      9.3.199.2
   domain          austin.ibm.com
   ```

   then type:

   ```
   nim -o define -t resolv_conf -a location=/exports/resolv.conf \
   -a server=master rconf1
   ```

   Prior to issuing the **bos_inst** operation, allocate this resource with other required and optional resources by typing:

   ```
   nim -o allocate -a spot=spot1 -a lpp_source=images1 \
   -a bosinst_data=bid1 -a resolv_conf=rconf1 client1
   ```

6. To allocate all resources applicable to standalone machines from the NIM resource group `res_grp1`, to the machine `mac1`, type:

   ```
   nim -o allocate -a group=res_grp1 mac1
   ```

deallocate

To deallocate an **lpp_source** named `images` from the standalone machine `client1`, type:

```
nim -o deallocate -a lpp_source=images client1
```

bos_inst

1. To install the machine `blowfish`, using the resources `spot1`, `images1`, `bosinst_data1`, and `rconf1`, first allocate the resources by typing:

   ```
   nim -o allocate -a spot=spot1 -a lpp_source=images1 \
   -a bosinst_data=bosinst_data1 -a resolv_conf=rconf1 blowfish
   ```

   Then, perform the BOS installation by typing:

   ```
   nim -o bos_inst blowfish
   ```

2. To install the machine `blowfish` while allocating the resources `spot1`, `images1`, `bosinst_data1`, and `rconf1` automatically when the **bos_inst** operation starts, type:

   ```
   nim -o bos_inst -a spot=spot1 -a lpp_source=images1 \
   -a bosinst_data=bosinst_data1 -a resolv_conf=rconf1 blowfish
   ```

3. To use the default resources when installing the machine `mac1`, type:

   ```
   nim -o bos_inst mac1
   ```

cust

1. To install a software product into a spot, spot1, such that the image for the installable option, adt, resides in the **lpp_source**, images, type:

   ```
   nim -o cust -a lpp_source=images -a filesets=adt spot1
   ```

2. To install a software product into a spot, spot1, such that the image for the installable option, adt, resides in the **lpp_source**, images, and the **installp_bundle**, bundle1, contains the name of the installable option, type:

   ```
   nim -o cust -a lpp_source=images -a installp_bundle=bundle1 spot1
   ```

3. To install a software product into a spot, spot1, such that the image for the installable option, adt, resides on a tape that is in the tape drive that is local to the machine where the spot resides, type:

   ```
   nim -o cust -a lpp_source=/dev/rmt0 -a filesets=adt spot1
   ```

4. To install a software product into a spot, spot1, such that the image for the installable option, adt, resides on a tape that is in the tape drive that is local to the machine where the spot resides, type:

   ```
   nim -o cust -a lpp_source=/dev/rmt0 -a filesets=adt spot1
   ```

5. To install all fileset updates associated with APAR IX12345, residing on the tape **/dev/rmt0** into spot1 and any diskless or dataless clients to which spot1 is currently allocated, type:

   ```
   nim -F -o cust -afixes=IX12345 -a lpp_source=/dev/rmt0 spot1
   ```

6. To update all software installed on the client Standalone1, with the latest updates in the **lpp_source** named updt_images, type:

   ```
   nim -o allocate -a lpp_source=updt_images Standalone1
   nim -o cust -afixes=update_all Standalone1
   ```

7. To install the machine catfish with the contents of the **installp_bundle** bundle1, first allocate the resources by typing:

   ```
   nim -o allocate -a installp_bundle=bundle1 \
   -a lpp_source=images1 catfish
   ```

   Then, perform the cust operation by typing:

   ```
   nim -o cust catfish
   ```

8. To update all software installed on the client Standalone1, with the latest updates in the **lpp_source** named updt_images, type:

   ```
   nim -o cust -a lpp_source=updt_images -a fixes=update_all \
   Standalone1
   ```

9. To install the machine catfish with the contents of the **installp_bundle** bundle1, while allocating this resource and the **lpp_source** images1 when the **cust** operation runs, type:

   ```
   nim -o cust -a installp_bundle=bundle1 -a lpp_source=images1 \
   catfish
   ```

maint

1. To deinstall the software products bos.INed and adt from a spot, spot1, type:

   ```
   nim -o maint -a installp_flags="-u" \
   -a filesets="bos.INed adt" spot1
   ```

2. To deinstall the options bos.INed and adt from a spot, spot1, such that the **installp_bundle**, bundle2, contains the names of the installable options, type:

   ```
   nim -o maint -a installp_flags="-u" \
   -a installp_bundle=bundle2 spot1
   ```

3. To cleanup from an interrupted software installation on a spot, spot1, type:

   ```
   nim -o maint -a installp_flags="-C" spot1
   ```

4. From the master, to deinstall the software products bos.INed and adt from a standalone machine, stand1, type:

   ```
   nim -o maint -a installp_flags="-u" \
   -a filesets="bos.INed adt" stand1
   ```

5. From the master, to clean up from an interrupted software installation on a standalone machine, stand1, type:

   ```
   nim -o maint -a installp_flags="-C" stand1
   ```

6. From the master, to deinstall the software products bos.INed and adt from a standalone machine, stand1, such that **installp_bundle**, bundle2, contains the names of the installable options, type:

   ```
   nim -o maint -a installp_flags="-u" \
   -a installp_bundle=bundle2 stand1
   ```

maint_boot

To enable the NIM standalone client, stand1, to boot in maintenance mode, type:

```
nim -o maint_boot stand1
```

This sets up the maintenance boot operation, but you must initiate the network boot locally from stand1.

dkls_init

1. To initialize the environment for a diskless workstation with the name of syzygy, using the resources spot1, root1, dump1, and paging1, first allocate the resources by typing:

   ```
   nim -o allocate -a spot=spot1 -a root=root1 -a dump=dump1 \
   -a paging=paging1 syzygy
   ```

   Then initialize the resources for the client machine by typing:

   ```
   nim -o dkls_init syzygy
   ```

2. To initialize the environment for a diskless workstation with the name of syzygy, type:

   ```
   nim -o dkls_init syzygy
   ```

3. To exclude the member named diskls2 from operations on the machine group DisklsMacs1, and then initialize the remaining members while allocating the diskless resources defined by the resource group named DisklsRes1, type the following two commands:

   ```
   nim -o select -a exclude=diskls2 DisklsMacs1
   nim -o dkls_init -a group=DisklsRes1 DisklsMacs1
   ```

4. To initialize the group of diskless machines defined by the machine group dtgrp1, while allocating the required and optional resources defined by the resource group dk_resgrp1, when the **dkls_init** operation runs, type:

   ```
   nim -o dkls_init -a group=dtgrp1 dk_resgrp1
   ```

dtls_init

1. To initialize the environment for a dataless workstation with the name of syzygy, using the resources spot1, root1, and dump1, first allocate the resources by typing:

   ```
   nim -o allocate -a spot=spot1 -a root=root1 -a dump=dump1 syzygy
   ```

   Then initialize the resources for the client machine by typing:

   ```
   nim -o dtls_init syzygy
   ```

2. To initialize the environment for a dataless workstation with the name of syzygy, type:

   ```
   nim -o dtls_init syzygy
   ```

3. To exclude the member named dataless1 from operations on the machine group DatalsMacs1, and then initialize the remaining members while allocating the dataless resources defined by the resource group named DatalsRes1, type the following two commands:

   ```
   nim -o select -a exclude=datals2 DatalsMacs1
   nim -o dtls_init -a group=DatalsMacs1 DatalsRes1
   ```

4. To initialize the group of dataless machines defined by the machine group DatalsMacs1, while allocating the required and optional resources defined by the resource group DatalsRes1, when the **dtls_init** operation runs, type:

   ```
   nim -o dtls_init -a group=DatalsMacs1 DatalsRes1
   ```

fix_query

To list information about fixes installed on client `Standalone1` for 20 APAR numbers, create the file `/tmp/apar.list` with one APAR number per line, as shown:

```
IX123435
IX54321
IX99999
...
```

then type:

```
nim -o define -t fix_bundle -alocation=/tmp/apar.list \
                -aserver=master fix_bun
nim -o allocate -a fix_bundle=fix_bun Standalone1
nim -o fix_query Standalone1
```

lppchk

1. To check fileset version and requisite consistency on the SPOT `spot1`, type:

   ```
   nim -o lppchk spot1
   ```

2. To verify the file checksums for all packages beginning with the name `bos` on NIM targets in the group of standalone machines `macgrp1`, and displaying detailed error information and updating the software database to match the actual file checksum when inconsistencies are found, type:

   ```
   nim -o lppchk -a lppchk_flags='-c -m3 -u' \
   -a filesets='bos*' macgrp1
   ```

   Because the **lppchk** operation runs in the background on group members by default, to view the output from the **lppchk** operation type:

   ```
   nim -o showlog -a log_type=lppchk macgrp1
   ```

showlog

To view the boot logs of the machines defined by the group `DisklsMacs1`, type:

```
nim -o showlog -a log_type=boot DisklsMacs1
```

showres

1. To show the contents of the config script `script1`, type:

   ```
   nim -o showres script1
   ```

2. To show the contents of the bosinst.data resource `bosinst_data1`, type:

   ```
   nim -o showres bosinst_data1
   ```

3. To list all the filesets in the lpp_source `lpp_source1`, type:

   ```
   nim -o showres lpp_source1
   ```

4. To list all the filesets in the lpp_source `lpp_source1` relative to what is currently installed on the machine `machine1`, type:

   ```
   nim -o showres -a reference=machine1 lpp_source1
   ```

5. To list user instructions for the `bos.INed` and `xlC.rte` filesets on the lpp_source `lpp_source1`, type:

   ```
   nim -o showres -a filesets="bos.INed xlC.rte" \
   -a installp_flags="qi" lpp_source1
   ```

6. To list all problems fixed by software on the lpp_source `lpp_source1`, use:

   ```
   nim -o showres -a instfix_flags="T" lpp_source1
   ```

AIX 4.3 or later Examples

alt_disk_install

1. To install a **mksysb** resource `all_devices_mysysb` to client `roundrock`, on hdisk4 and hdisk5, using the **image_data** resource `image_data_shrink`, with debug turned on, type:

```
nim -o alt_disk_install -a source=mksysb\
-a image_data=image_data_shrink\
-a debug=yes\
-a disk='hdisk4 hdisk5' roundrock
```

2. To clone a **rootvg** on client `austin` to hdisk2, but only run phase1 and phase2 (leaving the **/alt_inst** file systems mounted), type:

```
nim -o alt_disk_install -a source=rootvg\
-a disk='hdisk2'\
-a phase=12 austin
```

AIX 5.1 or later Examples

bos_inst

1. To install the machine blowfish and accept software license agreements, type:

```
nim -o bos_inst -a spot=spot1 -a lpp_source=images1 \
-a accept_licenses=yes -a resolv_conf=rconf1 blowfish
```

Files

/etc/niminfo	Contains variables used by NIM.

Related Information

The **lsnim** command, **nimclient** command, **nimconfig** command, **niminit** command.

The **.info** file.

---

nimadm Command

Purpose

Performs Alternate Disk Migration to a new version or release of AIX using NIM resources.

Syntax

Perform Alternate Disk Migration:

nimadm -l *lpp_source* **-c** *NIMClient* **-s** *SPOT* **-d** *TargetDisks* [**-a** *PreMigrationScript*] [**-b** *installp_bundle*] [**-z** *PostMigrationScript*] [**-e** *exclude_files*] [**-i** *image_data*] [**-m** *NFSMountOptions*] [**-o** *bosinst_data*] [**-P** *Phases*] [**-Y**] [**-F**] [**-D**] [**-E**] [**-V**] [{ **-B** | **-r** }]

Cleanup Alternate Disk Migration on client:

nimadm -C *NIMClient* **-s** *SPOT* [**-F**] [**-D**] [**-E**]

Wake-up Volume Group:

nimadm -W -c *NIMClient* **-s** *SPOT* **-d** *TargetDisks* [**-m** *NFSMountOptions*] [**-z** *PostMigrationScript*] [**-F**] [**-D**] [**-E**]

Put-to-sleep Volume Group:

nimadm -S -c *NIMClient* **-s** *SPOT* [**-F**] [**-D**] [**-E**]

Description

nimadm (Network Install Manager Alternate Disk Migration) is a utility that allows the system administrator to create a copy of **rootvg** to a free disk (or disks) and simultaneously migrate it to a new version or release level of AIX. **nimadm** uses NIM resources to perform this function.

There are several advantages to using nimadm over a conventional migration:

1. Reduced downtime. The migration is performed while the system is up and functioning normally. There is no requirement to boot from install media, and the majority of processing occurs on the NIM master.
2. **nimadm** facilitates quick recovery in the event of migration failure. Since **nimadm** uses **alt_disk_install** to create a copy of **rootvg**, all changes are performed to the copy (altinst_rootvg). In the even of serious migration installation failure, the failed migration is cleaned up and there is no need for the administrator to take further action. In the event of a problem with the new (migrated) level of AIX, the system can be quickly returned to the pre-migration operating system by booting from the original disk.
3. **nimadm** allows a high degree of flexibility and customization in the migration process. This is done with the use of optional NIM customization resources: image_data, bosinst_data, exclude_files, pre-migration script, installp_bundle, and post-migration script.

Please note that this document will only address subjects pertaining to **nimadm**. For complete coverage of alt_disk_install, NIM, migration, and other related install issues please refer to the latest editions of the following publications:

- ″Network Installation Management Guide and Reference″
- ″AIX Installation Guide″
- ″Migration Redbook″

nimadm Requirements

The **nimadm** requirements are:

1. Configured NIM master running AIX 5.1 or higher with AIX recommended maintenance level 5100-03 or higher.
2. The NIM master must have the same level of **bos.alt_disk_install.rte** installed in its **rootvg** and the SPOT which will be used to perform the migration. (Note: it is not necessary to install the **alt_disk_install** utilities on the client).
3. The selected **lpp_source** NIM resource, and selected SPOT NIM resource must match the AIX level to which you are migrating.
4. The NIM master should be at the same or higher AIX level then the level being migrated to.
5. The client (the system to be migrated) must be at AIX 4.3.3.0 or higher.
6. The client must have a disk (or disks) large enough to clone the **rootvg** and an additional 500 Megs (approximately) of free space for the migration. The total amount of required space will depend on original system configuration and **nimadm** customization.
7. The target client must be a registered with the master as a standalone NIM client (see the **niminit** command for more information).The nim master must be able to execute remote commands on the client using the **rshd** protocol.
8. The nim master must be able to execute remote commands on the client using the **rshd** protocol.
9. The NIM master and client must both have a minimum of 128 megabytess of RAM.
10. A reliable network, which can facilitate large amounts of NFS traffic, must exist between the NIM master and the client. The NIM master and client must be able to perform NFS mounts and read/write operations.
11. The client's hardware and software should support the AIX level that is being migrated to and meet all other conventional migration requirements.

Note: If you cannot meet requirements 1-10, you will need to perform a conventional migration. If you cannot meet requirement 11, then migration is not possible.

Attention: before performing a **nimadm** migration you will be required to agree to all software license agreements for software to be installed. You can do this by specifying the ″**-Y**″ flag as an argument to the **nimadm** command or setting the **ADM_ACCEPT_LICENSES** environment variable to ″yes″.

nimadm Limitations

The following limitations apply to **nimadm**:

1. If the client's **rootvg** has TCB turned on, you will need to either disable it (permanently) or perform a conventional migration. (This limitation exists because TCB needs to access file metadata which is not visible over NFS).
2. All NIM resources used by **nimadm** must be local to the NIM master.
3. Although there is almost no interference with the client's active **rootvg** during the migration, the client may experience minor performance decrease due to increased disk i/o, biod activity, and some CPU usage associated with **alt_disk_install** cloning.
4. NFS tuning may be required to optimize **nimadm** performance.

NIM resources used by nimadm:

SPOT resource (-s flag)

The NIM spot resource is required for all **nimadm** operations (migration, cleanup, wake-up, sleep). All **nimadm** and **alt_disk_install** utilities that will be used by the client are installed in this resource. It is not necessary to install **nimadm** software on the client. The NIM ″cust″ operation should be used to install the following filesets into the spot:

- Required: **bos.alt_disk_install.rte** (must match the NIM master's level).
- Optional message catalog: **bos.msg.***$LANG***.alt_disk_install.rte**

lpp_source resource (-l flag)

This NIM resource is the source of install images that will be used to migrate the system. It is required for nimadm migration operations. The **lpp_source** must contain all system images for the level being migrated to (check the lpp_source simages attribute in **lsnim -l lpp_source** output). It should also contain any optional installp images that need to be migrated.

pre-migration

This script resource that is run on the NIM master, but in the environment of the client's alt_inst file system that is mounted on the master (this is done by using the **chroot** command). This script is run before the migration begins.

post-migration

This script resource is similar to the **pre-migration** script, but it is executed after the migration is complete.

image_data

Specifies an **image_data** resource that is passed to **alt_disk_install** (as arguments to the **-i** flag). NIM will allocate and mount this resource on the client before calling **alt_disk_install**.

exclude_files

Specifies an **exclude_files** resource that is passed to **alt_disk_install** (as an argument to the -e flag). NIM will allocate and mount this resource on the client before calling **alt_disk_install**.

installp_bundle

This NIM resource specifies any additional software that **nimadm** will install after completing the migration.

bosinst_data

This NIM resource specifies various install settings that may be used by **nimadm**.

The nimadm Migration Process

nimadm performs migration in 12 phases. Each phase can be executed individually using the **-P** flag. The user should have a good understanding of the **nimadm** process before performing a migration in phases. The **nimadm** phases are as follows:

1. The master issues an The **alt_disk_install** command to the client which makes a copy of the **rootvg** to the target disks (coincidentally this is Phase 1 of the **alt_disk_install** process). In this phase **altinst_rootvg** (alternate rootvg) is created.
2. The master runs remote client commands to export all of the **/alt_inst** file systems to the master. The file systems are exported as read/write with root access to the master.
3. The master NFS mounts the file systems exported in Phase 2.
4. If a pre-migration script resource has been specified, it is executed at this time.
5. System configuration files are saved. Initial migration space is calculated and appropriate file system expansions are made. ″bos″ is restored and the device database is merged (similar to a conventional migration). All of the migration merge methods are executed and some miscellaneous processing takes place.
6. All system filesets are migrated using **installp**. Any required RPM images are also installed during this phase.
7. If a **post-migration** script resource has been specified, it is executed at this time.
8. **bosboot** is executed to create a client boot image, which is written out to the client's boot logical volume (**hd5**).
9. All mounts made on the master in phase 3 are removed.
10. All client exports created in phase 2 are removed.
11. **alt_disk_install** is called again (phase 3 of **alt_disk_install**) to make final adjustments and put **altinst_rootvg** to sleep. The bootlist is set to the target disk (unless the -B flag is used).
12. Cleanup is executed to end the migration. The client is rebooted, if the ″-r″ flag is specified.

Note: **nimadm** supports migrating several clients simultaneous.

nimadm Cleanup Operation

This operation, indicated with the ″**-C**″ flag, is designed to clean up after a failed migration that for some reason did not perform a cleanup it self. It can also be used to clear a previous migration in order to perform a new migration.

nimadm Wake-up and Sleep

After a migration completes, **nimadm** can be used to ″wake-up″ the migrated **altinst_rootvg** or the original **rootvg** (if booted from the migrated disk). The **nimadm** wake-up (**-W** flag) performs an **alt_disk_install** wakeup, NFS exports the **/alt_inst** file systems, and mounts them on the NIM master. The **nimadm** sleep function (**-S** flag) reverses the wake-up by unmounting the NIM master mounts, unexporting the **/alt_inst** file systems, and executing the **alt_disk_install** ″sleep″ function on the client.

Flags

-a *PreMigrationScript*	Specifies the pre-migration NIM script resource.
-b *installp_bundle*	Specifies the installp_bundle NIM resource.
-B	Specifies not running **bootlist** after **nimadm** migration. If set, then **-r** flag cannot be used.
-c *TargetDisks*	Specifies the NIM defined client which will be the target of this **nimadm** operation. This flag is required for all **nimadm** operations.
-C	Performs **nimadm** cleanup.
-d *TargetDisks*	Specifies the client target disk which will be used to create **altinst_rootvg** (the volume group that will be migrated).

-D	Sets **nimadm** into debug mode. This function should only be used to debug **nimadm** related problems and is not set by default.
-e *exclude_files*	Specifies the **exclude_files** NIM resource. This resource is used by the **alt_disk_install** command during Phase 1.
-E	Enters the **nimadm** debugger if a serious migration error occurs.
-F	Forces a client to unlock. Normally, **nimadm** locks a client to perform various operations. While the client is locked, other **nimadm** or NIM operations cannot be performed. This flag should ONLY be used in the unusual condition that a client is incorrectly locked (this can happen if for some reason **nimadm** could not call cleanup after a failure).
-i *image_data*	Specifies the **image_data** NIM resource. This resource is used by the **alt_disk_install** command during Phase 1 and 11.
-l *lpp_source*	Specifies the lpp_source NIM resource to be used for this **nimadm** operation. This flag is required for migration operations.
-m *NFSMountOptions*	Specifies arguments which will be passed to the mount command that **mounts** client resources on the master. This flag can be used to tune **nimadm** related NFS performance.
-o *bosinst_data*	Specifies **bosinst_data** NIM resource.
-P *Phase*	The phase to execute during this invocation of **nimadm**. If there is more then one phase, the phases should be separated by spaces or commas. Valid phases are 1 through 12.
-r	Specifies that the client should reboot after **nimadm** migration is complete.
-s *SPOT*	Specifies the **SPOT** NIM resource to be used for this **nimadm** operation. This flag is required for all **nimadm** operations.
-S	Performs the **nimadm** ″sleep″ function. This function should be executed to end a **nimadm** ″wake-up″.
-V	Turns on verbose output.
-W	Performs the **nimadm** ″wake-up″ function.
-Y	Agrees to required software license agreements for software to be installed.
-z *PostMigrationScript*	Specifies the post-migration NIM script resource.

Exit Status

0 All nimadm related operations completed successfully.

>0 An error occurred.

Security

Only the root user can execute **nimadm**.

Examples

1. To execute **nimadm** migration to target NIM client aix1, using NIM **SPOT** resource spot1, NIM **lpp_source** resource lpp1, and target disks hdisk1 & hdisk2. Note, the ″**-Y**″ flag agrees to all required software license agreements for software to be installed, enter the following:

   ```
   nimadm -c aix1 -s spot1 -l lpp1 -d "hdisk1 hdisk2" -Y
   ```

2. To execute the same operation as in the example above to hdisk2, and also run pre-migration script nimscript1 and post-migration script nimscript2, enter the following:

   ```
   nimadm -c aix1 -s spot1 -a nimscrip1 -z nimscript2 -l lpp1 -d hdisk1 -Y
   ```

3. To execute **nimadm** cleanup on client aix1, using NIM **SPOT** resource spot1, enter the following:

   ```
   nimadm -C -c aix1 -s spot1
   ```

Files

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

Related Information

The **lslpp** command, the **nim** command, the **lsnim** command, the **alt_disk_install** command, the **installp** command, the **chroot** command.

---

nimclient Command

Purpose

Allows Network Installation Management (NIM) operations to be performed from a NIM client.

Syntax

To Enable or Disable the NIM Master's Push Permissions

nimclient { **-p** } | { **-P** }

To List Information about the NIM Environment

nimclient -l *LsnimParameters*

To Set the Date and Time to That of the NIM Master

nimclient -d

To Perform a NIM Operation

nimclient -o *Operation* [**-a** *Attribute*=*Value*] ...

Description

The **nimclient** command is used by workstations that are NIM clients to pull NIM resources. This command can enable or disable the NIM master server's ability to initiate workstation installation and customization for the workstation. The **nimclient** command can be used to generate a list of available NIM resources or display the NIM resources that have already been allocated to the client. A limited set of NIM operations can also be performed by the **nimclient** command using the **-o** flag.

Flags

-a A*ttribute*=*Value*	Passes information to NIM operations. **From the master** Use the **lsnim -q** *Operation* **-t** *Type* command to get a list of valid attributes for a specific operation. **From the client** Use the **nimclient -l -q** *Operation* **-t** *Type* command to get a list of valid attributes for a specific operation.
-d	Sets the client's date and time to that of the master.

-l *Lsnim parameters*
Executes the **lsnim** command on the master using the **lsnim** parameters that you specify. All the parameters which you use with this option must adhere to the syntax rules of the **lsnim** command. Note that some **lsnim** syntax requires the use of a NIM object name. To find out what the NIM name is for your machine, look in the **/etc/niminfo** file.

-o *Operation*
Performs the specified operation. The possible operations are:

allocate
Allocates a resource for use.

bos_inst
Performs a BOS installation.

change
Changes an object's attributes.

check Checks the status of a NIM object.

cust Performs software customization.

deallocate
Deallocates a resource.

diag Enables a machine to boot a diagnostic image.

maint_boot
Enables a machine to boot in maintenance mode.

reset Resets an object's NIM state.

showres
Displays the contents of a NIM resource.

-p
Enables the NIM master to push commands.

-P
Removes the NIM master's permissions to push commands.

Note: The master can override this restriction by using the **-F** flag.

Security

Access Control: You must have root authority to run the **nimclient** command.

Examples

1. To list all the NIM resources which are available to this machine when its NIM name is `pluto`, enter:
   ```
   nimclient -l -L pluto
   ```
2. To list all the Shared Product Object Trees (SPOTs) which are available to this machine when its NIM name is `pluto`, enter:
   ```
   nimclient -l -L -t spot pluto
   ```
3. To list the operations which may be initiated from this machine, enter:
   ```
   nimclient -l -p -s pull_ops
   ```
4. To prevent the NIM master from running commands locally on the client, enter:
   ```
   nimclient -P
   ```
5. To allocate a spot resource named `myspot`, an **lpp_source** resource named `images`, and an **installp** bundle file name `dept_bundle`, enter:
   ```
   nimclient -o allocate -a spot=myspot -a lpp_source=images \
   -a installp_bundle=dept_bundle
   ```
6. To perform a base system installation after the required resources have been allocated, enter:
   ```
   nimclient -o bos_inst
   ```
7. From a standalone client, to allocate an **lpp_source** and install a software product such that the image for the installable option, `adt`, is contained in the **lpp_source**, `images`, enter:

```
nimclient -o allocate -a lpp_source=images
```

Then enter:

```
nimclient -o cust -a filesets="adt"
```

8. From a standalone client, to allocate an **lpp_source** and install a software product such that the image for the installable option, `adt`, is contained in the **lpp_source**, `images`, and the name of the installable option is contained in the **installp_bundle**, `bundle3`, enter:

```
nimclient -o allocate -a lpp_source=images \
-a installp_bundle=bundle3
```

Then enter:

```
nimclient -o cust
```

9. To install all fileset updates associated with APAR `IX12345`, residing in the **lpp_source** `updt_images`, enter:

```
nimclient -o allocate -a lpp_source=updt_images
nimclient -o cust -afixes=IX12345
```

10. To update all installed software on the client with the latest updates from the `updt_images` **lpp_source**, enter:

```
nimclient -o allocate -a lpp_source=updt_images
nimclient -o cust -afixes=update_all
```

11. To enable the system to boot in maintenance mode using a SPOT resource named `spot1`, enter:

```
nimclient -o maint_boot -a spot=spot1
```

This sets up the maintenance boot operation, but you must initiate the network boot locally.

12. To show the contents of the config script `script1`, enter:

```
nimclient -o showres -a resource=script1
```

13. To show the contents of the bosinst.data resource `bosinst_data1`, enter:

```
nimclient -o showres -a resource=bosinst_data1
```

14. To list all the filesets in the lpp_source `lpp_source1` relative to what is currently installed on the machine `machine1`, from the NIM client machine `machine1`**,** enter:

```
nimclient -o showres -a resource=lpp_source1
```

The **reference** attribute is automatically supplied by the **nimclient** command.

15. To list user instructions for the `bos.INed` and `xlC.rte` filesets on the lpp_source `lpp_source1`, enter:

```
nimclient -o showres -a filesets="bos.INed xlC.rte" \
-a resource=lpp_source1 -a installp_flags="qi"
```

16. To list all problems fixed by software on the lpp_source `lpp_source1`, use:

```
nimclient -o showres -a instfix_flags="T" -a resource=lpp_source1
```

17. To install the filesets listed in the NIM **installp_bundle** `client_bundle` using the **lpp_source** `client_images`, while automatically allocating these resources during the installation operation, enter:

```
nimclient -o cust -a installp_bundle=client_bundle \
-a lpp_source=client_images
```

18. To perform a base system installation while automatically allocating all applicable resources from the NIM resource group named `client_grp`, enter:

```
nimclient -o bos_inst -a group=client_grp
```

19. To perform a base system installation while automatically allocating all applicable resources from the NIM group defined as the default resource group on the master, enter:

```
nimclient -o bos_inst
```

Files

/etc/niminfo Contains variables used by NIM.

Related Information

The **lsnim** command, **nim** command, **nimconfig** command, **niminit** command.

The **.info** file.

---

nimconfig Command

Purpose

Initializes the Network Installation Management (NIM) master package.

Syntax

To Initialize the NIM master package

nimconfig -a pif_name=*Pif* **-a netname=***Objectname* [**-a master_port=***PortNumber*] [**-a platform=***Value*] [**-a registration_port=***PortNumber*] [**-a ring_speed=***Speed* | **-a cable_type=***CableType*]

To Rebuild the /etc/niminfo file:

nimconfig -r

Description

The **nimconfig** command initializes the NIM master package. You must initialize the package before any other NIM commands can be used. When you use the **-a** flag to supply the proper attributes, the **nimconfig** command initializes the NIM environment by performing the following tasks:

- Defines a network object specified by the *ObjectName* parameter to represent the network to which the NIM master's primary interface, specified by the *Pif* parameter, is connected.
- Completes the definition of the NIM master by connecting it to the newly defined network object.
- Defines a resource object to represent the network boot resource, which is managed automatically by NIM.
- Defines a resource object to represent the customization scripts that NIM automatically builds to perform customization.
- Starts the **NIM** communications daemon, **nimesis**.

Flags

-a Assigns the following attribute=value pairs:

pif_name=*Pif*
Designates the primary network interface for the NIM master. This value must be a logical interface name (such as tr0 or en0) is in the available state.

master_port=*PortNumber*
Specifies the port number of the **nimesis** daemon used for NIM client communication.

platform=*Value*
Specifies the platform. The supported platforms are:

rs6K Micro Channel-based, uniprocessor models for AIX 5.1 and earlier

rs6ksmp
Micro Channeled-based, symmetric multiprocessor models for AIX 5.1 and earlier

rspc PowerPC PCI bus-based, uniprocessor models for AIX 5.1 and earlier

rspcsmp
PowerPC PCI bus-based, symmetric multiprocessor models for AIX 5.1 and earlier

netname=*ObjectName*
Specifies the name you want the **nimconfig** command to use when creating the network object to represent the network to which the master's primary interface connects.

ring_speed=*Speed*
Speed in Mbps. When the **pif_name** refers to a token ring network, this value must be given. Acceptable values are:

4

16

cable_type=*CableType*
Specifies the ethernet cable type. When the **pif_name** refers to an ethernet network, this value must be given. Acceptable values are:

bnc

dix

N/A

registration_port=*PortNumber*
Specifies the port number used for NIM client registration.
Note: If you do not specify port numbers on the command line, the port numbers in the **/etc/services** file for NIM are used. If the **/etc/services** file does not contain entries for the NIM ports `nim` and `nimreg`, the default values of 1058 for **master_port** and 1059 for **registration_port** are used.

-r Rebuilds the **/etc/niminfo** file on the master using the information already exists in the NIM database. Note that if the **bos.sysmgt.nim.master** package has not been configured on this machine, this option will fail. This option is provided in case the **/etc/niminfo** file is accidentally removed by a user.

Security

Access Control: You must have root authority to run the **nimconfig** command.

Examples

1. To initialize the NIM environment using token ring and the default NIM ports for network communications, type:

   ```
   nimconfig -a pif_name=tr0 -a netname=net1 -a ring_speed=16
   ```

2. To initialize the NIM environment using ethernet and the default NIM ports, type:

```
nimconfig -a pif_name=en0 -a master_port=1058 \
-a netname = net2 -a cable_type=bnc
```

3. To rebuild the **/etc/niminfo** file on the NIM master when that machine has already been correctly configured as a master, type:

```
nimconfig -r
```

4. To initialize the NIM master using an ATM network interface, type:

```
nimconfig -a pif_name=at0 -a master_port=1058 -a netname=ATMnet
```

   Note: Because an interface to an ATM network does not currently support booting over the network, this operation will define a generic network object corresponding to the master's subnet.

5. To initialize the NIM environment using TCP/IP port 1060 for NIM client communications and TCP/IP port 1061 for NIM client registration, type:

```
nimconfig -a pif_name=tr0 -a netname=net2 -a master_port=1060 \
-a registration_port=1061 -a ring_speed=16
```

Files

/etc/niminfo	Contains variables used by NIM.

Related Information

The **lsnim** command, **nim** command, **nimclient** command, **niminit** command.

The **.info** file.

---

nimdef Command

Purpose

Defines Network Installation Management (NIM) clients from a stanza file.

Syntax

nimdef [**-p** | **-d** | **-c**] **-f** *Name*

Description

The **nimdef** command parses a definition stanza file to build the commands required to add NIM client definitions to the NIM environment.

The **nimdef** command can also create NIM networks and NIM machine groups automatically in the NIM environment to support the new client definitions.

> **Note:** Before using the **nimdef** command, you must configure the NIM master. (See Configuring the NIM Master and Creating Basic Installation Resources in *AIX 5L Version 5.2 Installation Guide and Reference* for more information.)

Client Definition File Rules

The format of the client definition file must comply with the following rules:

- After the stanza header, follow attribute lines of the form *Attribute = Value.*
- If you define an attribute value multiple times within the same stanza, only the last definition is used unless the attribute is **machine_group**. If you specify multiple **machine_group** attributes, all are applied to the machine definition.
- If you use an invalid attribute keyword, then that attribute definition is ignored.
- Each line of the file can have only one header or attribute definition.

- Only one stanza may exist in a definition file for each machine hostname.
- If the stanza header entry is the keyword **default**, this specifies to use it for the purpose of defining default values.
- You can specify a default value for any machine attribute except the machine hostname. If you do not specify an attribute for a machine but define a default value, then the default value is used.
- You can specify and change default values at any location in the definition file. After a default value is set, it applies to all definitions following it.
- To turn off a default value for all following machine definitions, set the attribute value to **nothing** in a default stanza.
- To turn off a default value for a single machine definition, set the attribute value to **nothing** in the machine stanza.
- You can include comments in a client definition file. Comments begin with the pound (**#**) character.
- When parsing the definition file for header/attribute keywords and values, tab characters and spaces are ignored.

Client Definition File Keywords

The client definition file uses the following keywords to specify machine attributes:

Required Attributes

cable_type Specifies the cable type of the machine. Required if **network_type** is **ent**.

gateway Specifies the hostname or IP address of the default gateway used by the machine. If the machine does not use a gateway, then specify the value **0** (zero) for this attribute.

machine_type Specifies the type of the machine: **standalone**, **diskless**, or **dataless**.

network_type Specifies the type of the machine's network adapter: **ent** or **tok**.

ring_speed Specifies the ring speed of the machine. Required if **network_type** is **tok**.

subnet_mask Specifies the subnet mask used by the machine.

Optional Attributes

nim_name Specifies the NIM name to use for a machine. Use this attribute if something other than the hostname is used for the NIM name. By default, the NIM name given to a machine is the hostname of the machine with any domain information stripped off. If you use non-unique hostnames in different domains, a conflict occurs because the same NIM name is used for both machines. In such an environment, define this attribute for the affected machine definitions.

platform Specifies the machine hardware platform. If you do not specify this attribute, default is **rs6k** through AIX 5.1 only.

net_adptr_name Specifies the name of the network adapter used by the machine (**tok0**, **ent0**, etc.).

netboot_kernel=*NetbootKernelType* Specifies the type of kernel to use when booting the client over the network. The **netboot_kernel** values are **up** or **mp**.

ipl_rom_emulation Specifies the device to use for IPL ROM emulation (**/dev/fd0**, **/dev/rmt0**, etc.).

primary_interface Specifies the hostname used for the original machine definition. Use this attribute if the current stanza is only to define an additional interface to a machine that is defined in the NIM environment.

master_gateway Specifies the gateway that the NIM master uses to reach this machine if this machine is on a different network. This attribute is not necessary if this machine is defined on a network that is already defined in the NIM environment, or if the NIM master network has a default gateway specified.

machine_group Specifies the group or groups to add the machine to when it is defined.

comments Specifies a comment to include in the machine definition. The comment string should be in double quotes (″).

Client Definition File Stanza Errors

A definition stanza is incorrect under any of the following conditions:

- The hostname used in the stanza header for the definition is unresolvable.
- A required attribute is missing.
- You specify an invalid value for an attribute.
- An attribute mismatch occurs. For example, you can not specify **network_type=tok** and **cable_type=bnc** in the same stanza.
- A group-type mismatch occurs. For example, you can not specify a group for a machine if the group includes standalone machines and you specify **machine_type=diskless**.
- Machine definitions occur multiple times for the same hostname.
- A machine definition occurs for a machine that is already defined in the NIM environment.
- The **primary_interface** value in a machine definition does not match the hostname of any defined machine or stanza definition.
- The **primary_interface** value in a machine definition matches the hostname of another machine definition, but that definition is incorrect.

Sample Client Definition File

These default values are for AIX 5.1 and earlier.

```
# Set default values.
default:
    machine_type  = standalone
    subnet_mask   = 255.255.240.0
    gateway       = gateway1
    network_type  = tok
    ring_speed    = 16
    platform      = rs6k
    machine_group = all_machines

# Define the machine "lab1"
# Take all defaults.
lab1:

# Define the machine "lab2"
# Take all defaults and specify 2 additional attributes.
# The machine "lab2" uses IPL ROM emulation, and will be added to
# the machine groups "all_machines" and "lab_machines".
lab2:
    ipl_rom_emulation = /dev/fd0
    machine_group     = lab_machines

# Define the machine "lab3"
# Take all defaults, but do not add the machine to the default
# group.
lab3:
    machine_group=

# Define the machine "lab4"
# Take all defaults, but do not add "lab4" to the default group
# "all_machines".
# Instead add it to the groups "lab_machines" and "new_machines".
lab4:
    machine_group =
    machine_group = lab_machines
    machine_group = new_machines

# Change the default "platform" attribute.
default:
    platform = rspc
```

```
# define the machine "test1"
# Take all defaults and include a comment.
test1:
    comments = "This machine is a test machine."
```

Flags

-c Generates commands from a client definition file. This flag processes the definition file and generates the commands to add the definitions. The commands are not invoked but displayed as a KSH script that you can redirect to a file and invoke at a later time.

-d Defines machines from a client definition file. This flag processes the definition file and invokes the commands to add the definitions to the NIM environment.

-f *Name* Specifies the name of the client definition file.

-p Displays a preview of the client definition file. This flag processes the definition file but does not add machines to the NIM environment. Displays the following:

All complete and valid NIM definition stanzas.

All additional interfaces that will be defined for machines.

All invalid definitions stanzas and the reason for failure.

All new machine groups and the members to add.

All existing machine groups and the members to add.

All network definitions to add to the NIM environment.

The commands to invoke to add each new machine.

The commands to invoke to add each additional machine interface.

The commands to invoke to create new machine groups and add their members.

The commands to invoke to add new members to existing machine groups.

> **Note:** We recommend that you specify the **-p** flag on a client definition file to verify that all stanzas are correct before using it for adding machines.

Exit Status

This command returns the following exit values:

0 Successful completion.

!0 An error occurred.

Security

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

Auditing Events: N/A

Examples

1. To preview the client definition file **client.defs**, enter:
   ```
   nimdef -p -f client.defs
   ```
2. To add the NIM clients described in the client definition file **client.defs**, enter:
   ```
   nimdef -d -f client.defs
   ```

3. To create a kshell script called **client.add** to add the NIM clients described in the client definition file **client.defs**, enter:

```
nimdef -c -f client.defs > client.add
```

Files

/usr/sbin/nimdef	Contains the **nimdef** daemon/command.

Related Information

The **lsnim** command, **nim** command, **nimclient** command, **nimconfig** command.

---

niminit Command

Purpose

Configures the Network Installation Management (NIM) client package.

Syntax

To Configure the NIM Client Package

niminit{**-a name=***Name* **-a pif_name=***Pif* **-a master=***Hostname*} [**-a master_port=***PortNumber*] [**-a registration_port=***PortNumber*] [**-a cable_type=***Type* | **-a ring_speed=***Speed*] [**-a iplrom_emu=***Device*] [**-a platform=***PlatformType*] [**-a netboot_kernel=***NetbootKernelType*] [**-a adpt_add=***AdapterAddress*]

To Rebuild the /etc/niminfo File

niminit {**-a name=***Name* **-a master=***Hostname* **-a master_port=***PortNumber*}

Description

The **niminit** command configures the NIM client package. This must be done before the **nimclient** command can be used. When the required attributes are supplied to the **niminit** command, a new machine object will be created to represent the machine where the **niminit** command is being executed. When the **niminit** command completes successfully, the machine will be able to participate in the NIM environment.

After the NIM client package has been successfully configured, the **niminit** command can be run again to rebuild the **/etc/niminfo** on the client. The **/etc/niminfo** file is used by the **nimclient** command and must be rebuilt if it is accidentally removed by a user.

Flags

-a Specifies up to five different attributes for the **niminit** command. All of the following attribute=value pairs are preceded by the **-a** flag:

name=*Name*	Specifies the name that NIM will use to identify the workstation. This value is required.
pif_name=*Pif*	Defines the name of the network interface for all NIM communications. This value is required.
master=*Hostname*	Specifies the hostname of the NIM master. The client must have the ability to resolve this hostname to an Internet Protocol (IP) address. This value is required.
master_port=*PortNumber*	Specifies the port number of the **nimesis** daemon used for NIM communications.

cable_type=*CableType*
Specifies the ethernet cable type. When the **pif_name** refers to an ethernet network, this value must be given. Acceptable values are: **bnc**, **dix**, and **N/A**.

ring_speed=*Speed*
Speed in Mbps. When the **pif_name** refers to a token ring network, this value must be given. Acceptable values are: **4** and **16**.

iplrom_emu=*Device*
Specifies a device that contains a ROM emulation image. This image is required for models that do not have internal support for booting via network interface.

platform=*PlatformType*
Specifies the platform that corresponds to the client's machine type. If this attribute is not specified, the default, **rs6k**, will be used. The supported platforms are:

rs6k Micro Channel-based, uniprocessor models for AIX 5.1 and earlier

rs6ksmp
Micro Channel-based, symmetric multiprocessor models for AIX 5.1 and earlier

rspc PowerPC PCI bus-based, uniprocessor machines for AIX 5.1 and earlier

rspcsmp
PowerPC PCI bus-based, symmetric multiprocessor machines for AIX 5.1 and earlier

adpt_add=*AdapterAddress*
Specifies the hardware address that corresponds to the network adapter.

registration_port=*PortNumber*
Specifies the port number used for NIM client registration.

Notes:

1. If you do not specify port numbers on the command line, the port numbers in the **/etc/services** file for NIM is used. If the **/etc/services** file does not contain entries for the NIM ports `nim` and `nimreg`, the default values of 1058 for **master_port** and 1059 for **registration_port** are used.
2. The values used for **master_port** and **registration_port** should match the values used by the NIM master. To display the values used by the NIM master, run the command **lsnim -l master** on the NIM master.

netboot_kernel= *NetbootKernelType*
Specifies the type of kernel to use when booting the client over the network. The **netboot_kernel** values are:

up Kernel for uniprocessor machines

mp Kernel for multiprocessor machines

The default is **up**.

Security

Access Control: You must have root authority to run the **niminit** command.

Examples

1. To configure the NIM client package on a machine that has a BOOTP-enabled IPL ROM such that it will be known as `scuba` in the NIM environment, using `en0` as its primary interface and an ethernet cable type of `bnc`, and specifying that it communicates with the NIM master using the master's hostname of `manta` and the default NIM ports located in **/etc/services** for network install communications, type:

```
niminit -a name=scuba -a pif_name=en0 -a cable_type=bnc \
-a master=manta
```

2. To rebuild the **/etc/niminfo** file when it has accidentally been removed by a user, using a hostname of `superman` for the master's hostname and a port number of `1058`, type:

   ```
   niminit -a name=robin -a master=superman -a master_port=1058
   ```

3. To configure the NIM client package for AIX 5.1 and earlier on a machine that is a PowerPC PCI bus-based, uniprocessor system that has a BOOTP-enabled IPL ROM such that it will be known as `starfish` in the NIM environment, using `en0` as its primary interface and an Ethernet cable type of `dix`, and specifying that it communicates with the NIM master using the master's host name of `whale` and a port number of `1058`, type:

   ```
   niminit -a name=starfish -a pif_name=en0 -a cable_type=dix \
   -a master=whale -a master_port=1058 -a platform=rspc
   ```

4. To configure the NIM client, on a machine to be known as `bluefish` in the NIM environment, using `at0` as its primary interface and specifying that it communicates with the NIM master using the master's host name `redfish` and a port number of `1058`, type:

   ```
   niminit -a name=bluefish -a pif_name=at0 -a master=redfish \
   -a master_port=1058
   ```

   Note: Because an interface to an ATM network does not currently support booting over the network, this operation will define a machine object on the NIM master if a Generic network object corresponding to the client's subnet is already defined.

5. To configure the NIM client for AIX 5.1 and earlier on a machine that is a PowerPC PCI bus-based, symmetric multiprocessor system that has a BOOTP-enabled IPL ROM such that it will be it will be known as `jellyfish` in the NIM environment, using `en0` as its primary interface and an Ethernet cable type of `dix`, and specifying that it communicates with the NIM master using the master's host name of `whale` and a port number of `1058`, type:

   ```
   niminit -a name=jellyfish -a pif_name=en0 -a cable_type=dix \
   -a master=whale -a master_port=1058 -a platform=rspcsmp
   ```

6. To configure the NIM client package on a machine that will use an IPL ROM emulation in device `/dev/fd0`, such that it will be known as `octopus` in the NIM environment and uses `tr0` as its primary interface and a ring speed of `16`, and communicates with the NIM master using the master's hostname of `dolphin` and a port number of `1700` for client communications and `1701` for client registration, type:

   ```
   niminit -a iplrom_emu=/dev/fd0 -a name=octopus -a pif_name=tr0 \
   -a ring_speed=16 -a master=dolphin -a master_port=1700 \
   -a registration_port=1701
   ```

Files

/etc/niminfo	Contains variables used by NIM.

Related Information

The **lsnim** command, **nim** command, **nimclient** command, **nimconfig** command.

The **.info** file.

---

nis_cachemgr Daemon

Purpose

Starts the NIS+ cache manager daemon.

Syntax

nis_cachemgr [**-i**] [**-n** [**-v**]

Description

The **nis_cachemgr** daemon maintains a cache of the NIS+ directory objects. The cache contains location information necessary to contact the NIS+ servers that serve the various directories in the name space. This includes transport addresses, information neeeded to authenticate the server, and a time to live field which gives a hint on how long the directory object can be cached. The cache helps to improve the performance of the clients that are traversing the NIS+ name space. The **nis_cachemgr** daemon should be running on all the machines that are using NIS+. However, it is not required that the **nis_cachemgr** daemon be running for NIS+ requests to be serviced.

The cache maintained by this daemon is shared by all the processes that access NIS+ on that machine. The cache is maintained in a file that is memory mapped by all the processes. On start up, the **nis_cachemgr** daemon initializes the cache from the cold start file and preserves unexpired entries that already exist in the cache file. Thus, the cache survives machine reboots.

The **nis_cachemgr** daemon is normally started from a system startup script. The **nis_cachemgr** daemon makes NIS+ requests under the NIS+ principal name of the host on which it runs. Before running the **nis_cachemgr** daemon, security credentials for the host should be added to the cred.org_dir table in the host's domain using the **nisaddcred** command. Credentials of type DES are needed if the NIS+ service is operating at security level 2 (see the **rpc.nisd** command). Additionally, `keylogin -r` needs to be done on the machine.

> **Attention:** If the host principal does not have the proper security credentials in the cred.org_dir table for its domain, then running this daemon without the **-n** insecure mode flag may significantly degrade the performance of processes issuing NIS+ requests.

Flags

-i Forces the **nis_cachemgr** daemon to ignore the previous cache file and reinitialize the cache from just the cold start file. By default, the cache manager initializes itself from both the cold start file and the old cache file, thereby maintaining the entries in the cache across machine reboots.

-n Runs the **nis_cachemgr** daemon in an insecure mode. By default, before adding a directory object to the shared cache on the request of another process on the machine, it checks the encrypted signature on the request to make sure that the directory object is a valid one and is sent by an authorized server. In this mode, the **nis_cachemgr** daemon adds the directory object to the shared cache without making this check.

-v Sets verbose mode. In this mode, the **nis_cachemgr** daemon logs not only errors and warnings but also additional status messages. The additional messages are logged using **syslog** with a priority of LOG_INFO.

Diagnostics

The **nis_cachemgr** daemon logs error messages and warnings using **syslog**. Error messages are logged to the DAEMON facility with a priority of LOG_ERR and warning messages with a priority of LOG_WARNING. Additional status messages can be obtained using the **-v** flag.

Files

/var/nis/NIS_SHARED_DIRCACHE	Contains the shared cache file
/var/nis/NIS_COLD_START	Contains the coldstart file
/etc/init.d/rpc	Contains initialization scripts for NIS+

Related Information

The **keylogin** command, **nisaddcred** command, **nisinit** command, **nisshowcache** command.

The **rpc.nisd** daemon .

nisaddcred Command

Purpose

Creates NIS+ credential information.

Syntax

nisaddcred [**-p** *principal*] [**-P** *nis_principal*] [**-l** *login_password*] *auth_type* [*domain_name*]

nisaddcred -r [*nis_principal*] [*domain_name*]

Description

The **nisaddcred** command is used to create security credentials for NIS+ principals. NIS+ credentials serve two purposes. The first is to provide authentication information to various services; the second is to map the authentication service name into a NIS+ principal name.

When the **nisaddcred** command is run, these credentials get created and stored in a table named cred.org_dir in the default NIS+ domain. If *domain_name* is specified, the entries are stored in the cred.org_dir of the specified domain. The specified domain must either be the one to which you belong or one in which you are authenticated and authorized to create credentials, that is, a subdomain. Credentials of normal users must be stored in the same domain as their passwords.

It is simpler to add credentials using the **nisclient** command because it obtains the required information itself. The **nispopulate** command is used for bulk updates and can also be used to add credentials for entries in the hosts and the passwd NIS+ tables.

NIS+ principal names are used in specifying clients that have access rights to NIS+ objects. Various other services can also implement access control based on these principal names.

The cred.org_dir table is organized as follows :

cname	**auth_type**	**auth_name**	**public_data**	**private_data**
user1.foo.com.	LOCAL	2990	10,102,44	
user1.foo.com.	DES	unix.2990@foo.com	098...819	3b8...ab2

The **cname** column contains a canonical representation of the NIS+ principal name. By convention, this name is the login name of a user or the host name of a machine followed by a dot ('.') followed by the fully qualified home domain of that principal. For users, the home domain is defined to be the domain where their DES credentials are kept. For hosts, their home domain is defined to be the domain name returned by the **domainname** command executed on that host.

There are two types of *auth_type* entries in the cred.org_dir table. Those with authentication type LOCAL and those with authentication type DES. *auth_type*, specified on the command line in upper or lower case, should be either local or des.

Entries of type LOCAL are used by the NIS+ service to determine the correspondence between fully qualified NIS+ principal names and users identified by UIDs in the domain containing the cred.org_dir table. This correspondence is required when associating requests made using the AUTH_SYS RPC authentication flavor to a NIS+ principal name. It is also required for mapping a UID in one domain to its fully qualified NIS+ principal name whose home domain may be elsewhere. The principal's credentials for any authentication flavor may then be sought for within the cred.org_dir table in the principal's home domain (extracted from the principal name). The same NIS+ principal may have LOCAL credential entries in more than one domain. Only users, and not machines, have LOCAL credentials. In their home domain, users of NIS+ should have both types of credentials.

The *auth_name* associated with the LOCAL type entry is a UID that is valid for the principal in the domain containing the cred.org_dir table. This may differ from that in the principal's home domain. The public information stored in *public_data* for this type contains a list of GIDs for groups in which the user is a member. The GIDs also apply to the domain in which the table resides. There is no private data associated with this type. Neither a UID nor a principal name should appear more than once among the LOCAL entries in any one cred.org_dir table.

The DES *auth_type* is used for Secure RPC authentication.

The authentication name associated with the DES *auth_type* is a Secure RPC netname. A Secure RPC netname has the form unix.id@*domain*.com, where *domain* must be the same as the domain of the principal. For principals that are users, the id must be the UID of the principal in the principal's home domain. For principals that are hosts, the id is the host's name. In Secure RPC, processes running under effective UID 0 (root) are identified with the host principal. Unlike LOCAL, there cannot be more than one DES credential entry for one NIS+ principal in the NIS+ namespace.

The public information in an entry of authentication type DES is the public key for the principal. The private information in this entry is the private key of the principal encrypted by the principal's network password.

User clients of NIS+ should have credentials of both types in their home domain. In addition, a principal must have a LOCAL entry in the cred.org_dir table of each domain from which the principal wishes to make authenticated requests. A client of NIS+ that makes a request from a domain in which it does not have a LOCAL entry is unable to acquire DES credentials. A NIS+ service running at security level 2 or higher considers such users unauthenticated and assign them the name **nobody** for determining access rights.

This command can only be run by those NIS+ principals who are authorized to add or delete the entries in the cred table.

If credentials are being added for the caller itself, **nisaddcred** automatically performs a keylogin for the caller.

You can list the cred entries for a particular principal with **nismatch**.

Flags

-l *login_password*
: Use the *login_password* specified as the password to encrypt the secret key for the credential entry. This overrides the prompting for a password from the shell. This flag is intended for administration scripts only. Prompting guarantees not only that no one can see your password on the command line using the **ps** command, but it also checks to make sure you have not made any mistakes.

  Note: *login_password* does not have to be the user's password; but, if it is, it simplifies logging in.

-p *principal*
: Specifies the name of the principal as defined by the naming rules for that specific mechanism. For example, LOCAL credential names are supplied with this flag by including a string specifying a UID. For DES credentials, the name should be a Secure RPC netname of the form unix.id@*domain*.com, as described earlier. If the **-p** flag is not specified, the *auth_name* field is constructed from the effective UID of the current process and the name of the local domain.

-P *nis_principal*
: Use the NIS+ principal name *nis_principal*. This flag should be used when creating LOCAL or DES credentials for users whose home domain is different than the local machine's default domain. Whenever the **-P** flag is not specified, **nisaddcred** constructs a principal name for the entry as follows. When it is not creating an entry of type LOCAL, **nisaddcred** calls **nis_local_principal**, which looks for an existing LOCAL entry for the effective UID of the current process in the cred.org_dir table and uses the associated principal name for the new entry. When creating an entry of authentication type LOCAL, **nisaddcred** constructs a default NIS+ principal name by taking the login name of the effective UID for its own process and appending to it a dot ('.') followed by the local machine's default domain. If the caller is a superuser, the machine name is used instead of the login name.

-r [*nis_principal*]
: Remove all credentials associated with the principal *nis_principal* from the cred.org_dir table. This flag can be used when removing a client or user from the system. If *nis_principal* is not specified, the default is to remove credentials for the current *user*. If *domain_name* is not specified, the operation is executed in the default NIS+ domain.

Exit Status

This command returns the following exit values:

0	Success
1	Failure

Examples

1. To add the LOCAL and DES credentials for some user, user1, with a UID of 2990, who is an NIS+ user principal in the some.domain.com. NIS+ domain, enter:

   ```
   nisaddcred -p 2990 -P user1.some.domain.com. local
   ```

   Credentials are always added in the cred.org_dir table in the domain where **nisaddcred** is run, unless *domain_name* is specified as the last parameter on the command line. If credentials are being added from the domain server for its clients, then *domain_name* should be specified. The caller should have adequate permissions to create entries in the cred.org_dir table.
2. To add a DES credential for the same user, the system administrator can enter:

   ```
   nisaddcred -p unix.2990@some.domain.com -P user1.some.domain.com. des
   ```

   DES credentials can be added only after the LOCAL credentials have been added. The secure RPC netname does not end with a dot ('.') while the NIS+ principal name (specified with the **-P** flag) does. This command should be executed from a machine in the same domain as is the user.
3. To add a machine's DES credentials in the same domain, enter:

   ```
   nisaddcred -p unix.foo@some.domain.com -P foo.some.domain.com. des
   ```

   No LOCAL credentials are needed in this case.
4. To add a NIS+ workstation's principal DES credential, enter:

   ```
   nisaddcred -p unix.host1@sub.some.domain.com \
   -P newhost.sub.some.domain.com. des sub.some.domain.com.
   ```

   This format is particularly useful if you are running this command from a server that is in a higher domain than sub.some.domain.com. Without the last option for domain name, **nisaddcred** would fail because it would attempt to use the default domain of some.domain.com.
5. To add DES credentials without being prompted for the root login password, enter:

   ```
   nisaddcred -p unix.2990@some.domain.com -P user1.some.domain.com. -l
   login_password des
   ```

Related Commands

The **chkey** command, **domainname** command, **keylogin** command, **niscat** command, **nischmod** command, **nischown** command, **nisclient** command, **nismatch** command, **nispopulate** command, **ps** command.

---

nisaddent Command

Purpose

Creates NIS+ tables from corresponding **/etc** files or NIS maps.

Syntax

nisaddent [**-D** *defaults*] [**-P**] [**-a**] [**-r**] [**-v**] [**-t** *table*] *type* [*nisdomain*]

nisaddent [**-D** *defaults*] [**-P**] [**-a**] [**-p**] [**-r**] [**-m**] [**-v**] **-f** *file* [**-t** *table*] *type* [*nisdomain*]

nisaddent [**-D** *defaults*] [**-P**] [**-a**] [**-r**] [**-m**] [**-v**] [**-t** *table*] **-y** *ypdomain* [**-Y** *map*] *type* [*nisdomain*]

nisaddent -d [**-A**] [**-M**] [**-q**] [**-t** *table*] *type* [*nisdomain*]

Description

The **nisaddent** command creates entries in NIS+ tables from their corresponding **/etc** files and NIS maps. This operation is customized for each of the standard tables that are used in the administration of systems. The *type* argument specifies the type of the data being processed. Legal values for this type are one of **aliases**, **bootparams**, **ethers**, **group**, **hosts**, **netid**, **netmasks**, **networks**, **passwd**, **protocols**, **publickey**, **rpc**, **services**, **shadow**, or **timezone** for the standard tables or **key-value** for a generic two-column (key, value) table. For a site specific table, which is not of **key-value** type, you can use **nistbladm** to administer it.

The NIS+ tables should have already been created by **nistbladm**, **nissetup**, or **nisserver**.

It is easier to use **nispopulate** instead of **nisaddent** to populate the system tables.

By default, **nisaddent** reads from the standard input and adds this data to the NIS+ table associated with the *type* specified on the command line. An alternate NIS+ table may be specified with the **-t** flag. For type **key-value**, a table specification is required.

> **Note:** The *data* type can be different than the table name (**-t**). For example, the automounter tables have **key-value** as the table type.

Although, there is a *shadow* data type, there is no corresponding *shadow* table. Both the shadow and the passwd data is stored in the **passwd** table itself.

Files may be processed using the **-f** flag, and NIS version 2 (YP) maps may be processed using the **-y** flag. The **-m** flag is not available when reading data from standard input.

When a *ypdomain* is specified, the **nisaddent** command takes its input from the **dbm** files for the appropriate NIS map (**mail.aliases**, **bootparams**, **ethers.byaddr**, **group.byname**, **hosts.byaddr**, **netid.byname**, **netmasks.byaddr**, **networks.byname**, **passwd.byname**, **protocols.byname**, **publickey.byname**, **rpc.bynumber**, **services.byname**, or **timezone.byname**). An alternate NIS map may be specified with the **-Y** flag. For type **key-value**, a map specification is required. The map must be in the **/var/yp/***ypdomain* directory on the local machine.

> **Note:** *ypdomain* is case sensitive. The **ypxfr** command can be used to get the NIS maps.

If a *nisdomain* is specified, **nisaddent** operates on the NIS+ table in that NIS+ domain, otherwise the default domain is used.

In terms of performance, loading up the tables is fastest when done through the **dbm** files (**y**).

Flags

-a Adds the file or map to the NIS+ table without deleting any existing entries. This flag is the default. This mode only propagates additions and modifications, not deletions.

-A Specifies that the data within the table and all of the data in tables in the initial table's concatenation path be returned.

-d Dumps the NIS+ table to the standard output in the appropriate format for the given *type*. For tables of type **key-value**, use **niscat** instead. To dump the credential table, dump the **publickey** and the **netid** types.

-D *defaults* Specifies a different set of defaults to be used during this operation. The *defaults* string is a series of tokens separated by colons. These tokens represent the default values to be used for the generic object properties. All of the legal tokens are described below:

ttl=*time*
Sets the default time to live for objects that are created by this command. The value *time* is specified in the format as defined by the **nischttl** command. The default is 12 hours.

owner=*ownername*
Specifies that the NIS+ principal *ownername* should own the created object. The default for this value is the principal who is executing the command.

group=*groupname*
Specifies that the group *groupname* should be the group owner for the object that is created. The default is **NULL**.

access= *rights*
Specifies the set of access rights that are to be granted for the given object. The value *rights* is specified in the format as defined by the **nischmod** command. The default is **——rmcdr—r—-**.

-f *file* Specifies that *file* should be used as the source of input (instead of the standard input).

-m Merges the file or map with the NIS+ table. This is the most efficient way to bring a NIS+ table up to date with a file or NIS map when there are only a small number of changes. This flag adds entries that are not already in the database, modifies entries that already exist (if changed), and deletes any entries that are not in the source. Use the **-m** flag whenever the database is large and replicated and the map being loaded differs only in a few entries. This flag reduces the number of update messages that have to be sent to the replicas. Also see the **-r** flag.

-M Specifies that lookups should be sent to the master server. This guarantees that the most up-to-date information is seen at the possible expense that the master server may be busy or that it may be made busy by this operation.

-p Processes the password field when loading password information from a file. By default, the password field is ignored because it is usually not valid (the actual password appears in a shadow file).

-P Specifies that lookups should follow the concatenation path of a table if the initial search is unsuccessful.

-q	Dumps tables in ″quick″ mode. The default method for dumping tables processes each entry individually. For some tables (for example, hosts), multiple entries must be combined into a single line, so extra requests to the server must be made. In ″quick″ mode, all of the entries for a table are retrieved in one call to the server, so the table can be dumped more quickly. However, for large tables, there is a chance that the process will run out of virtual memory and the table will not be dumped.
-r	Replaces the file or map in the existing NIS+ table by first deleting any existing entries and then add the entries from the source (**/etc** files or NIS+ maps). This flag has the same effect as the **-m** flag. The use of this flag is strongly discouraged due to its adverse impact on performance, unless there are a large number of changes.
-t *table*	Specifies that *table* should be the NIS+ table for this operation. This should be a relative name as compared to your default domain or the *domainname* if it has been specified.
-v	Sets verbose mode.
-y *ypdomain*	Uses the **dbm** files for the appropriate NIS map, from the NIS domain *ypdomain*, as the source of input. The files are expected to be on the local machine in the **/var/yp/***ypdomain* directory. If the machine is not an NIS server, use the **ypxfr** command to get a copy of the **dbm** files for the appropriate map.
-Y *map*	Use the **dbm** files for *map* as the source of input.

Environment

NIS_DEFAULTS	This variable contains a default string that overrides the NIS+ standard defaults. If the **-D** flag is used, those values will then override both the **NIS_DEFAULTS** variable and the standard defaults. To avoid security accidents, the access rights in the **NIS_DEFAULTS** variable are ignored for the **passwd** table but access rights specified with the **-D** flag are used.
NIS_PATH	If this variable is set and neither the *nisdomain* nor the *table* are fully qualified, each directory specified in **NIS_PATH** will be searched until the table is found (see the **nisdefaults** command).

Exit Status

This command returns the following exit values:

0	Success
1	Failure caused by an error other than parsing
2	A parsing error occurred on an entry. A parsing error does not cause termination; the invalid entries are simply skipped.

Examples

1. To add the contents of **/etc/passwd** to the **passwd.org_dir** table, enter:
   ```
   cat /etc/passwd | nisaddent passwd
   ```
2. To add the shadow information, enter:
   ```
   cat /etc/shadow | nisaddent shadow
   ```

   The table type is shadow, not passwd, even though the actual information is stored in the **passwd** table.
3. To replace the **hosts.org_dir** table with the contents of **/etc/hosts** (in verbose mode), enter:

```
nisaddent -rv -f /etc/hosts hosts
```

4. To merge the **passwd** map from **yypdomain** with the **passwd.org_dir.nisdomain** table (in verbose mode), enter:

```
nisaddent -mv -y myypdomain passwd nisdomain
```

   This example assumes that the **/var/yp/myypdomain** directory contains the **yppasswd** map.

5. To merge the **auto.master** map from **myypdomain** with the **auto_master.org_dir** table, enter:

```
nisaddent -m -y myypdomain -Y auto.master -t auto_master.org_dir key-value
```

6. To dump the **hosts.org_dir** table, enter:

```
nisaddent -d hosts
```

Related Information

The **niscat** command, **nischmod** command, **nisdefaults** command, **nispopulate** command, **nisserver** command, **nissetup** command, **nistbladm** command, **passwd** command, **ypxfr** command.

---

niscat Command

Purpose

Displays the contents of an NIS+ table.

Syntax

niscat [**-A**] [**-h**] [**-L**] [**-M**] [**-v**] *tablename*

niscat [**-A**] [**-L**] [**-M**] [**-P**] **-o** *name*

Description

In the first syntax, the **niscat** command displays the contents of the NIS+ tables named by *tablename*. In the second syntax, it displays the internal representation of the NIS+ objects named by *name*.

Flags

-A Displays the data within the table and all of the data in tables in the initial table's concatenation path.

-h Displays the header line prior to displaying the table. The header consists of the # character followed by the name of each column. The column names are separated by the table separator character.

-L Follows links. When this flag is specified if *tablename* or *name* names a LINK type object, the link is followed and the object or table named by the link is displayed.

-M Specifies that the request should be sent to the master server of the named data. This guarantees that the most up-to-date information is seen at the possible expense of increasing the load on the master server and increasing the possibility of the NIS+ server being unavailable or busy for updates.

-o Displays the internal representation of the named NIS+ objects. If *name* is an indexed name, then each of the matching entry objects is displayed. This flag is used to display access rights and other attributes of individual columns.

-P Follows concatenation path. This flag specifies that the request should follow the concatenation path of a table if the initial search is unsuccessful. This flag is only useful when using an indexed name for *name* and the **-o** flag.

-v Displays binary data directly. This flag displays columns containing binary data on the standard output. Without this flag, binary data is displayed as the string `*BINARY*`.

Environment

NIS_PATH If this variable is set and the NIS+ name is not fully qualified, each directory specified will be searched until the object is found (see the **nisdefaults** command).

Exit Status

This command returns the following exit values:

0	Success
1	Failure

Examples

1. To display the contents of the host's table, type:

   ```
   niscat -h hosts.org_dir
   # cname name addr comment
   client1 client1 129.144.201.100 Joe Smith
   crunchy crunchy 129.144.201.44 Jane Smith
   crunchy softy 129.144.201.44
   ```

   The string *NP* is returned in those fields where the user has insufficient access rights.
2. To display the passwd.org_dir on the standard output, type:

   ```
   niscat passwd.org_dir
   ```
3. To display the contents of table frodo and the contents of all tables in its concatenation path, type:

   ```
   niscat -A frodo
   ```
4. To display the entries in the table groups.org_dir as NIS+ objects, type:

   ```
   niscat -o '[ ]groups.org_dir'
   ```

   The brackets are protected from the shell by single quotation marks.
5. To display the table object of the passwd.org_dir table, type:

   ```
   niscat -o passwd.org_dir
   ```

   The previous example displays the passwd table object and not the passwd table. The table object includes information such as the number of columns, column type, searchable or not searchable separator, access rights, and other defaults.
6. To display the directory object for org_dir, which includes information such as the access rights and replica information, type:

   ```
   niscat -o org_dir
   ```

Related Information

The **nistbladm** command, **nisdefaults** command, **nismatch** command.

---

nischgrp Command

Purpose

Changes the group owner of a NIS+ object.

Syntax

nischgrp [**-A**] [**-f**] [**-L**] [**-P**] *group name*

Description

The **nischgrp** command changes the group owner of the NIS+ objects or entries specified by *name* to the specified NIS+ *group*. Entries are specified using indexed names. If *group* is not a fully qualified NIS+ group name, it is resolved using the directory search path. For additional information, see the **nisdefaults** command.

The only restriction on changing an object's group owner is that you must have modify permissions for the object.

This command will fail if the master NIS+ server is not running.

The NIS+ server will check the validity of the group name prior to effecting the modification.

Flags

-A	Modifies all entries in all tables in the concatenation path that match the search criterion specified in *name*. This flag implies the **-P** flag.
-f	Forces the operation and fails silently if it does not succeed.
-L	Follows links and changes the group owner of the linked object or entries rather than the group owner of the link itself.
-P	Follows the concatenation path within a named table. This flag is valid when either *name* is an indexed name or the **-L** flag is also specified and the named object is a link pointing to entries.

Environment

NIS_PATH	If this variable is set and the NIS+ name is not fully qualified, each directory specified will be searched until the object is found (see the **nisdefaults** command).

Exit Status

This command returns the following exit values:

0	Success
1	Failure

Examples

1. To change the group owner of an object to a group in a different domain, enter:
   ```
   nischgrp newgroup.remote.domain. object
   ```
2. To change the group owner of an object to a group in the local domain, enter:
   ```
   nischgrp my-buds object
   ```
3. To change the group owner for a password entry, enter:
   ```
   nischgrp admins '[uid=99],passwd.org_dir'
   ```
   admins is a NIS+ group in the same domain.
4. To change the group owner of the object or entries pointed to by a link, enter:
   ```
   nischgrp -L my-buds linkname
   ```
5. To change the group owner of all entries in the **hobbies** table, enter:
   ```
   nischgrp my-buds '[],hobbies'
   ```

Related Information

The **nischmod** command, **nischown** command, **nisdefaults** command, **nisgrpadm** command.

---

nischmod Command

Purpose

Changes the access rights on a NIS+ object.

Syntax

nischmod [**-A**] [**-f**] [**-L**] [**-P**] *mode name*...

Description

The **nischmod** command changes the access rights (mode) of the NIS+ objects or entries specified by *name* to *mode*. Entries are specified using indexed names. Only principals with modify access to an object may change its mode.

mode has the following form:

rights [,*rights*]...

rights has the form:

[*who*] *op permission* [*op permission*]...

who is a combination of:

n	Nobody's permissions
o	Owner's permissions
g	Group's permissions
w	World's permissions
a	All, or **owg**

If *who* is omitted, the default is **a**.

op is one of:

+	Grants the *permission*
-	Revokes the *permission*
=	Sets the permissions explicitly

permission is any combination of:

r	Read
m	Modify
c	Create
d	Destroy

Flags

-A	Modifies all entries in all tables in the concatenation path that match the search criteria specified in *name*. This flag implies the **-P** flag.

-f	Forces the operation and fails silently if it does not succeed.
-L	Follows links and changes the permission of the linked object or entries rather than the permission of the link itself.
-P	Follows the concatenation path within a named table. This flag is only applicable when either *name* is an indexed name or the **-L** flag is also specified and the named object is a link pointing to an entry.

Environment

NIS_PATH	If this variable is set and the NIS+ name is not fully qualified, each directory specified will be searched until the object is found (see the **nisdefaults** command).

Exit Status

This command returns the following exit values:

0	Success
1	Failure

Examples

1. To give everyone read access to an object. (that is, access for owner, group, and all), enter:
   ```
   nischmod a+r object
   ```
2. To deny create and modify privileges to **group** and unauthenticated clients (**nobody**), enter:
   ```
   nischmod gn-cm object
   ```
3. To set a complex set of permissions for an object, enter:
   ```
   nischmod o=rmcd,g=rm,w=rc,n=r object
   ```
4. To set the permissions of an entry in the password table so that the group owner can modify them, enter:
   ```
   nischmod g+m '[uid=55],passwd.org_dir'
   ```
5. To change the permissions of a linked object, enter:
   ```
   nischmod -L w+mr linkname
   ```

Related Information

The **chmod** command, **nischgrp** command, **nischown** command, **nisdefaults** command.

---

nischown Command

Purpose

Changes the owner of one or more NIS+ objects or entries.

Syntax

nischown [**-A**] [**-f**] [**-L**] [**-P**] *owner name*...

Description

The **nischown** command changes the owner of the NIS+ objects or entries specified by *name* to *owner*. Entries are specified using indexed names. If *owner* is not a fully qualified NIS+ principal name (see the **nisaddcred** command), the default domain (see the **nisdefaults** command) will be appended to it.

The only restriction on changing an object's owner is that you must have modify permissions for the object.

> **Note:** If you are the current owner of an object and you change ownership, you may not be able to regain ownership unless you have modify access to the new object.

The command fails if the master NIS+ server is not running.

The NIS+ server will check the validity of the name before making the modification.

Flags

-A	Modifies all entries in all tables in the concatenation path that match the search criteria specified in *name*. It implies the **-P** flag.
-f	Forces the operation and fails silently if it does not succeed.
-L	Follows links and changes the owner of the linked object or entries rather than the owner of the link itself.
-P	Follows the concatenation path within a named table. This flag is only meaningful when either *name* is an indexed name or the **-L** flag is also specified and the named object is a link pointing to entries.

Environment

NIS_PATH	If this variable is set and the NIS+ name is not fully qualified, each directory specified will be searched until the object is found (see the **nisdefaults** command).

Exit Status

This command returns the following exit values:

0	Success
1	Failure

Examples

1. To change the owner of an object to a principal in a different domain, enter:
   ```
   nischown bob.remote.domain. object
   ```
2. To change the owner of an object to a principal in the local domain, enter:
   ```
   nischown skippy object
   ```
3. To change the owner of an entry in the passwd table, enter:
   ```
   nischown bob.remote.domain. '[uid=99],passwd.org_dir'
   ```
4. To change the object or entries pointed to by a link, enter:
   ```
   nischown -L skippy linkname
   ```

Related Information

The **nisaddcred** command, **nischgrp** command, **nischttl** command, **nischmod** command, **nisdefaults** command.

nischttl Command

Purpose

The **nischttl** command changes the *time-to-live* value of objects or entries in the namespace.

Syntax

To Change the Time-to-Live Value of Objects

nischttl [**-A**] [**-L**] [**-P**] [*time-to-live*] [*object-name*]

To Change the Time-to-Live Value of Entries

nischttl [*time-to-live*] [*column=value*,...] [*table-name*] [**-A**] [**-L**] [**-P**]

Note: Where time-to-live is expressed as:

- **Number of seconds.** A number with no letter is interpreted as a number of seconds. Thus, **1234** for TTL would be interpreted as 1234 seconds. A number followed by the letter **s** is also interpreted as a number of seconds. Thus, **987s** for TTL would be interpreted as 987 seconds. When seconds are specified in combination with days, hours, or minutes, you must use the letter **s** to identify the seconds value.
- **Number of minutes.** A number followed by the letter **m** is interpreted as a number of minutes. Thus, **90m** for TTL would be interpreted as 90 minutes.
- **Number of hours.** A number followed by the letter **h** is interpreted as a number of hours. Thus, **9h** for TTL would be interpreted as 9 hours.
- **Number of days.** A number followed by the letter **d** is interpreted as a number of days. Thus, **7d** for TTL would be interpreted as 7 days.

Note: These values may be used in combination. For example, a TTL value of **4d3h2m1s** would specify a time to live of four days, three hours, two minutes, and one second.

Description

This *time-to-live* value is used by the cache manager to determine when to expire a cache entry. You can specify the *time-to-live* in total number of seconds or in a combination of days, hours, minutes, and seconds. The *time-to-live* values you assign objects or entries should depend on the stability of the object. If an object is prone to frequent change, give it a low time-to-live value. If it is steady, give it a high one. A high time-to-live is a week; a low one is less than a minute. Password entries should have *time-to-live* values of about 12 hours to accommodate one password change per day. Entries in tables that don't change much, such as those in the RPC table, can have values of several weeks.

Notes

1. To change the *time-to-live* of an object, you must have modify rights to that object. To change the *time-to-live* of a table entry, you must have modify rights to the table, entry, or columns you wish to modify.
2. To display the current *time-to-live* value of an object or table entry, use the **nisdefaults -t** command, described in Administering NIS+ Access Rights.

Flags

-A Apply the change to all the entries that match the column=value specifications that you supply.
-L Follow links and apply the change to the linked object or entry rather than the link itself.
-P Follow the path until there is one entry that satisfies the condition.

Examples

Changing the Time-to-Live of an Object

1. To change the *time-to-live* of an object, type the **nischttl** command with the *time-to-live* value and the object-name. You can add the **-L** command to extend the change to linked objects.

   nischttl -L *time-to-live* object-name

2. You can specify the *time-to-live* in seconds by typing the number of seconds. Or, you can specify a combination of days, hours, minutes, and seconds by using the suffixes **s, m, h**, and **d** to indicate the number of seconds, minutes, days, and hours. For example:

TTL of 86400 seconds	client% **nischttl 86400 sales.wiz.com.**
TTL of 24 hours	client% **nischttl 24h sales.wiz.com.**
TTL of 2 days, 1 hour, 1 minute, and 1 second	client% **nischttl 2d1h1m1s sales.wiz.com.**

3. The first two commands change the *time-to-live* of the sales.wiz.com. directory to 86,400 seconds, or 24 hours. The third command changes the *time-to-live* of all the entries in a hosts table to 2 days, 1 hour, 1 minute, and 1 second.

Changing the Time-to-Live of a Table Entry

1. To change the *time-to-live* of entries, use the indexed entry format. You can use any of the options, **-A**, **-L**, or **-P**.

   nischttl [-ALP] *time-to-live* **[**column=value**,...],**
   table-name

2. These examples are similar to those above, but they change the value of table entries instead of objects:

   ```
   client% nischttl 86400 '[uid=99],passwd.org_dir.wiz.com.'
   client% nischttl 24h `[uid=99],passwd.org_dir.wiz.com.'
   client% nischttl 2d1h1m1s `[name=fred],hosts.org_dir.wiz.com'
   ```

   NoteC shell users should use quotes to prevent the shell from interpreting the square bracket ([) as a metacharacter.

Related Information

The **defaults** command.

---

nisclient Command

Purpose

Initializes NIS+ credentials for NIS+ principals.

Syntax

Add DES Credentials for NIS+ Principals

nisclient -c [**-x**] [**-o**] [**-v**] [**-l** *network_password*] [**-d** *NIS+_domain*] *client_name*...

Initialize a NIS+ Client Machine

nisclient -i [**-x**] [**-v**] **-h** *NIS+_server_host* [**-a** *NIS+_server_addr*] [**-d** *NIS+_domain*] [**-S** *0* | *2*]

Initialize a NIS+ User

nisclient -u [**-x**] [**-v**]

Restore Network Service Environment

nisclient -r [**-x**]

Description

The **nisclient** command can be used to:

- Create NIS+ credentials for hosts and users
- Initialize NIS+ hosts and users
- Restore the network service environment

NIS+ credentials are used to provide authentication information of NIS+ clients to NIS+ service.

Use the first syntax (**-c**) to create individual NIS+ credentials for hosts or users. You must be logged in as a NIS+ principal in the domain for which you are creating the new credentials. You must also have write permission to the local credential table. The *client_name* argument accepts any valid host or user name in the NIS+ domain (for example, the *client_name* must exist in the hosts or passwd table). The **nisclient** command verifies each *client_name* against both the **host** and **passwd** tables, then adds the proper NIS+ credentials for hosts or users.

> **Note:** If you are creating NIS+ credentials outside your local domain, the host or user must exist in the **host** or **passwd** tables in both the local and remote domains.

By default, **nisclient** will not overwrite existing entries in the credential table for the hosts and users specified. To overwrite, use the **-o** flag. After the credentials have been created, **nisclient** will print the command that must be executed on the client machine to initialize the host or the user. The **-c** flag requires a network password for the client which is used to encrypt the secret key for the client. You can either specify it on the command line with the **-l** flag or the script will prompt you for it. You can change this network password later with either the **nispasswd** or **chkey** command.

The **-c** flag is not intended to be used to create NIS+ credentials for all users and hosts that are defined in the **passwd** and **hosts** tables. To define credentials for all users and hosts, use the **nispopulate** command.

Use the second syntax (**-i**) to initialize a NIS+ client machine. The **-i** flag can be used to convert machines to use NIS+ or to change the machine's domainname. You must be logged in as superuser on the machine that is to become a NIS+ client. Your administrator must have already created the NIS+ credential for this host by using the **nisclient -c** or **nispopulate -C** command. You will need the network password your administrator created. The **nisclient** command will prompt you for the network password to decrypt your secret key and then for this machine's root login password to generate a new set of secret/public keys. If the NIS+ credential was created by your administrator using **nisclient -c**, then you can simply use the initialization command that was printed by the **nisclient** script to initialize this host instead of typing it manually.

To initialize an unauthenticated NIS+ client machine, use the **-i** flag with **-S** *0*. With these flags, the **nisclient -i** flag will not ask for any passwords.

During the client initialization process, files that are being modified are backed up as *files*.no_nisplus. The files that are usually modified during a client initialization are: **/etc/defaultdomain**, **/etc/nsswitch.conf**, **/etc/inet/hosts**, and, if it exists, **/var/nis/NIS_COLD_START**.

> **Note:** A file will not be saved if a backup file already exists.

The **-i** flag does not set up a NIS+ client to resolve hostnames using DNS. Refer to the DNS documentation for information on setting up DNS. (See information on the resolv.conf) file format.

It is not necessary to initialize either NIS+ root master servers or machines that were installed as NIS+ clients.

Use the third syntax (**-u**) to initialize a NIS+ user. You must be logged in as the user on a NIS+ client machine in the domain where your NIS+ credentials have been created. Your administrator should have already created the NIS+ credential for your username using the **nisclient** or **nispopulate** command. You will need the network password your administrator used to create the NIS+ credential for your username. The **nisclient** command will prompt you for this network password to decrypt your secret key and then for your login password to generate a new set of secret/public keys.

Use the fourth syntax (**-r**) to restore the network service environment to whatever you were using before **nisclient -i** was executed. You must be logged in as superuser on the machine that is to be restored. The restore will only work if the machine was initialized with **nisclient -i** because it uses the backup files created by the **-i** flag.

Reboot the machine after initializing a machine or restoring the network service.

Flags

-a *NIS+_server_addr*	Specifies the IP address for the NIS+ server. This flag is used only with the **-i** flag.
-c	Adds DES credentials for NIS+ principals.
-d *NIS+_domain*	Specifies the NIS+ domain where the credential should be created when used in conjunction with the **-c** flag. It specifies the name for the new NIS+ domain when used in conjunction with the **-i** flag. The default is your current domainname.
-h *NIS+_server_host*	Specifies the NIS+ server's hostname. This flag is used only with the **-i** flag.
-i	Initializes a NIS+ client machine.
-l *network_password*	Specifies the network password for the clients. This flag is used only with the **-c** flag. If this flag is not specified, the script will prompt you for the network password.
-o	Overwrite existing credential entries. The default is not to overwrite. This is used only with the **-c** flag.
-r	Restores the network service environment.
-S *0* | *2*	Specifies the authentication level for the NIS+ client. Level 0 is for unauthenticated clients and level 2 is for authenticated (DES) clients. The default is to set up with level 2 authentication. This is used only with the **-i** flag. The **nisclient** command always uses level 2 authentication (DES) for both **-c** and **-u** flags. There is no need to run **nisclient** with **-u** and **-c** for level 0 authentication.
-u	Initializes a NIS+ user.
-v	Runs the script in verbose mode.
-x	Turns the echo mode on. The script just prints the commands that it would have executed. Note that the commands are not actually executed. The default is off.

Examples

1. To add the DES credential for host *dilbert* and user *fred* in the local domain, enter:
   ```
   nisclient -c dilbert fred
   ```
2. To add the DES credential for host *dilbert* and user *fred* in domain `xyz.ibm.com.`, enter:
   ```
   nisclient -c -d xyz.ibm.com. dilbert fred
   ```
3. To initialize host *dilbert* as a NIS+ client in domain `xyz.ibm.com.` where *nisplus_server* is a server for the domain `xyz.ibm.com.`, enter:
   ```
   nisclient -i -h nisplus_server -d xyz.ibm.com.
   ```

The script will prompt you for the IP address of *nisplus_server* if the server is not found in the **/etc/hosts** file. The **-d** flag is needed only if your current domain name is different from the new domain name.

4. To initialize host *dilbert* as an unauthenticated NIS+ client in domain `xyz.ibm.com.` where *nisplus_server* is a server for the domain `xyz.ibm.com.`, enter:

   ```
   nisclient -i -S 0 -h nisplus_server -d xyz.ibm.com. -a 129.140.44.1
   ```

5. To initialize user *fred* as a NIS+ principal, log in as user *fred* on a NIS+ client machine by entering:

   ```
   nisclient -u
   ```

Files

/var/nis/NIS_COLD_START	This file contains a list of servers, their transport addresses, and their Secure RPC public keys that serve the machines default domain.
/etc/defaultdomain	The system default domainname
/etc/nsswitch.conf	Configuration file for the name-service switch
/etc/inet/hosts	Local host name database

Related Information

The **chkey** command, **keylogin** command, **keyserv** command, **nisaddcred** command, **nisinit** command, **nispopulate** command.

resolv.conf file format.

---

nisdefaults Command

Purpose

Displays the seven default values currently active in the namespace.

Syntax

nisdefaults [**-d** *domain*] [**-g** *group*] [**-h** *host*] [**-p** *principal*] [**-r** *access_rights*] [**-s** *search_path*] [**-t** *time_to_live*] [**-a** *all(terse)*] [**-v** *verbose*]

Description

The **nisdefaults** command displays the seven default values currently active in the namespace. To display NIS+ defaults the default values are either:

- Preset values supplied by the NIS+ software
- The defaults specified in the **NIS_DEFAULTS** environment variable (if you have **NIS_DEFAULTS** values set)

Any object that you create on this machine will automatically acquire these default values unless you override them with the **-D** flag of the command you are using to create the object.

Setting Default Security Values

This section describes how to perform tasks related to the **nisdefaults** command, the **NIS_DEFAULTS** environment variable, and the **-D** flag. The **NIS_DEFAULTS** environment variable specifies the following default values:

- Owner
- Group
- Access rights

- Time-to-live

The values that you set in the **NIS_DEFAULTS** environment variable are the default values applied to all NIS+ objects that you create using that shell (unless overridden by using the **-D** flag with the command that creates the object).

You can specify the default values (owner, group, access rights, and time-to-live) specified with the **NIS_DEFAULTS** environment variable. After you set the value of `NIS_DEFAULTS`, every object you create from that shell will acquire those defaults, unless you override them by using the **-D** flag when you invoke a command.

Displaying the Value of NIS_DEFAULTS

You can check the setting of an environment variable by using the **echo** command, as shown in the following example:

```
client% echo $NIS_DEFAULTS
owner=butler:group=gamblers:access=o+rmcd
```

You can also display a general list of the NIS+ defaults active in the namespace by using the **nisdefaults** command.

Changing Defaults

You can change the default access rights, owner, and group, by changing the value of the **NIS_DEFAULTS** environment variable. Use the environment command that is appropriate for your shell (**setenv** for **csh** or **$NIS_DEFAULTS=**, **export** for **sh** and **ksh**) with the following arguments:

- **access=**right, where right are the access rights using the formats described in Specifying Access Rights in Commands.
- **owner=**name, where name is the user name of the owner.
- **group=**group, where group is the name of the default group.

You can combine two or more arguments into one line separated by colons:

owner=*principal-name***:group=***group-name*

Changing Defaults—Examples

Tasks	Examples
This command grants owner read access as the default access right.	`client%` **setenv NIS_DEFAULTS access=o+r**
This command sets the default owner to be the user abe whose home domain is Wiz.com.	`client%` **setenv NIS_DEFAULTS owner=abe.wiz.com.**
This command combines the first two examples on one code line.	`client%` **setenv NIS_DEFAULTS access=o+r:owner=abe.wiz.com.**

All objects and entries created from the shell in which you changed the defaults will have the new values you specified. You cannot specify default settings for a table column or entry; the columns and entries simply inherit the defaults of the table.

Resetting the Value of NIS_DEFAULTS

You can reset the NIS_DEFAULTS variable to its original values, by typing the name of the variable without arguments, using the format appropriate to your shell:

For C shell:

```
client# unsetenv NIS_DEFAULTS
```

For Bourne or Korn shell:

```
client$ NIS_DEFAULTS=; export NIS_DEFAULTS
```

Flags

-d *domain*	Displays the home domain of the workstation from which the command was entered. Displays the value of **/etc/defaultdomin** environment variable.
-g *group*	Displays the group that would be assigned to the next object created from this shell. Displays the value of **NIS_GROUP** environment variable.
-h *host*	Displays the workstation's host name. Displays the value of **uname -n** environment variable.
-p *principal*	Displays the fully qualified user name or host name of the NIS+ principal who entered the **nisdefaults** command. Displays the value of **gethostbyname()** environment variable.
-r *access_rights*	Displays the access rights that will be assigned to the next object or entry created from this shell. Format: **——rmcdr—-r—-**. Displays the value of **NIS_DEFAULTS** environment variable.
-s *search_path*	Displays the syntax of the search path, which indicate the domains that NIS+ will search through when looking for information. Displays the value of the **NIS_PATH** environment variable if it is set. Displays the value of **NIS_PATH** environment variable.
-t *time_to_live*	Displays the time-to-live that will be assigned to the next object created from this shell. The default is 12 hours. Displays the value of the **NIS_DEFAULTS** environment variable.
-a *all (terse)*	Displays all seven defaults in terse format. Displays the value of the environment variable.
-v *verbose*	Display specified values in verbose mode. Displays the value of the environment variable.

Note: You can use these options to display all default values or any subset of them.

Examples

1. To display all values in verbose format, type the **nisdefaults** command without arguments.

```
master% nisdefaults
Principal Name : topadmin.wiz.com.
Domain Name    : Wiz.com.
Host Name      : rootmaster.wiz.com.
Group Name     : salesboss
Access Rights  : ----rmcdr---r---
Time to live   : 12:00:00:00:00
Search Path    : Wiz.com.
```

2. To display all values in terse format, add the **-a** option.
3. To display a subset of the values, use the appropriate options. The values are displayed in terse mode. For example, to display the rights and search path defaults in terse mode, type:

```
rootmaster% nisdefaults -rs
----rmcdr---r---
Wiz.com.
```

4. To display a subset of the values in verbose mode, add the **-v** flag.

---

niserror Command

Purpose

Displays NIS+ error messages.

Syntax

niserror *error-num*

Description

The **niserror** command prints the NIS+ error associated with status value *error-num* on the standard output. It is used by shell scripts to translate NIS+ error numbers that are returned into text messages.

Examples

To print the error associated with the error number 20, enter:

```
niserror 20
Not Found, no such name
```

nisgrep Command

Purpose

Utility for searching NIS+ tables.

Syntax

nisgrep [**-A**] [**-c**] [**-h**] [**-M**] [**-o**] [**-P**] [**-s** [*sep*]] [**-v**]

Descripton

The **nisgrep** command can be used to search NIS+ tables. The command **nisgrep** differs from the **nismatch** command in its ability to accept regular expressions **keypat** for the search criteria rather than simple text matches.

Because **nisgrep** uses a callback function, it is not constrained to searching only those columns that are specifically made searchable at the time of table creation. This makes it more flexible, but slower, than **nismatch**.

In **nismatch**, the server does the searching; whereas in **nisgrep**, the server returns all the readable entries and then the client does the pattern-matching.

In both commands, the parameter **tablename** is the NIS+ name of the table to be searched. If only one key or key pattern is specified without the column name, then it is applied searching the first column. Specific named columns can be searched by using the **colname=key** syntax. When multiple columns are searched, only entries that match in all columns are returned. This is the equivalent of a logical join operation.

nismatch accepts an additional form of search criteria, **indexedname**, which is a NIS+ indexed name of the form:

```
colname=value, . . . ],tablename
```

Flags

-A	All data. Return the data within the table and all of the data in tables in the initial table's concatenation path.
-c	Print only a count of the number of entries that matched the search criteria.
-h	Display a header line before the matching entries that contains the names of the table's columns.
-M	Master server only. Send the lookup to the master server of the named data. This guarantees that the most up to date information is seen at the possible expense that the master server may be busy.
-o	Display the internal representation of the matching NIS+ object(s).
-P	Follow concatenation path. Specify that the lookup should follow the concatenation path of a table if the initial search is unsuccessful.
-s *sep*	This option specifies the character to use to separate the table columns. If no character is specified, the default separator for the table is used.
-v	Verbose. Do not suppress the output of binary data when displaying matching entries. Without this option binary data is displayed as the string * **BINARY** * .

Return Values

0	Successfully matches some entries.
1	Successfully searches the table and no matches are found.

2 An error condition occurs. An error message is also printed.

Examples

This example searches a table named **passwd** in the **org_dir** subdirectory of the **zotz.com.** domain. It returns the entry that has the username of **skippy**. In this example, all the work is done on the server.

```
example% nismatch name=skippy passwd.org_dir.zotz.com.
```

This example is similar to the one above except that it uses **nisgrep** to find all users in the table named **passwd** that are using either **ksh** or **csh**.

```
example% nisgrep 'shell=[ck]sh' passwd.org_dir.zotz.com.
```

NIS_PATH If this variable is set, and the NIS+ table name is not fully qualified, each directory specified will be searched until the table is found (see **nisdefaults**).

Related Information

The **niscat** command, **nisdefaults** command, **nisls** command, and **nistbladm** command.

---

nisgrpadm Command

Purpose

Creates, deletes, and performs miscellaneous administration operations on NIS+ groups.

Note: To use **nisgrpadm**, you must have access rights appropriate for the operation.

Syntax

To Create or Delete a Group or to List the Members

nisgrpadm [**-c** *group_name.domain_name*] [[**-d**] [**-l** *group_name*]]

To Add or Remove Members or Determine if They Belong to the Group

nisgrpadm [[**-a**] [**-r**] [**-t**] *group_name*]]

Note: A member can be any combination of the six membership types.

Description

The **nisgrpadm** command has two main forms, one for working with groups and one for working with group members.

All operations except create (**-c**) accept a partially qualified *group-names*. However, even for the **-c** flag, **nisgrpadm** will not accept the use of *groups_dir* in the *group-name* argument.

Flags

To Create or Delete a Group or to List the Members

-c *group_name.domain_name*	Creates an NIS+ group. You must have create rights to the *groups_dir* directory of the group's domain.
-d *group_name*	Deletes an NIS+ group. You must have destroy rights to the *groups_dir* directory in the group's domain.
-l *group_name*	Lists the members of an NIS+ group. You must have read rights to the group object.

To Add or Remove Members or Determine if They Belong to the Group

-a *group_name* Adds members to an NIS+ group. You must have modify rights to the group object.
-r *group_name* Removes members from an NIS+ group. You must have modify rights to the group object.
-t *group_name* Find out whether an NIS+ principal is a member of a particular NIS+ group. You must have read access to the group object.

Related Information
The **nisdefaults** command.

---

nisinit Command

Purpose
Initializes a workstation to be a NIS+ client.

Syntax

To Initialize a Client
nisinit [**-c** [**-k** *key_domain*] [**-C** *coldstart* | **-H** *host*| **-B**]]

To Initialize a Root Master Server
nisinit -r

To Initialize a Parent Server
[**-p Y**| **D**| **N** *parent_domain_host*...]

Description
The **nisinit** command initializes a workstation to be an NIS+ client. As with the **rpc.nisd** command, you don't need any access rights to use the **nisinit** command, but you should be aware of its prerequisites and related tasks.

Flags

-c Initializes the machine to be a NIS+ client. There are three initialization options available: initialize by *coldstart*, initialize by *hostname*, and initialize by broadcast. The most secure mechanism is to initialize from a trusted *coldstart* file. The second option is to initialize using a *hostname* that you specify as a trusted host. The third method is to initialize by broadcast and it is the least secure method.

-C*coldstart*
Causes the file *coldstart* to be used as a prototype coldstart file when initializing a NIS+ client. This *coldstart* file can be copied from a machine that is already a client of the NIS+ namespace. For maximum security, an administrator can encrypt and encode (with **uuencode**(1C)) the *coldstart* file and mail it to an administrator bringing up a new machine. The new administrator would then decode (with **uudecode**), decrypt, and then use this file with the **nisinit** command to initialize the machine as an NIS+ client. If the *coldstart* file is from another client in the same domain, the **nisinit** command may be safely skipped and the file copied into the **/var/nis** directory as **/var/nis/NIS_COLD_START**.

-H*hostname*
Specifies that the host *hostname* should be contacted as a trusted NIS+ server. The **nisinit** command will iterate over each transport in the NETPATH environment variable and attempt to contact **rpcbind** on that machine. This hostname must be reachable from the client without the name service running. For IP networks this means that there must be an entry in **/etc/hosts** for this host when **nisinit** is invoked.

-B Specifies that the **nisinit** command should use an IP broadcast to locate a NIS+ server on the local subnet. Any machine that is running the NIS+ service may answer. No guarantees are made that the server that answers is a server of the organization's namespace. If this flag is used, it is advisable to check with your system administrator that the server and domain served are valid. The binding information can be written to the standard output using the **nisshowcache** command.

Note: nisinit -c will just enable navigation of the NIS+ namespace from this client. To make NIS+ your name service, modify the file **/etc/nsswitch.conf** to reflect that.

-k*key_domain*
Specifies the domain where root's credentials are stored. If it is not specified, then the system default domain is assumed. This domain name is used to create the **/var/nis/NIS_COLD_START** file.

-pY|D|N*parent_domain host...*
Initialize on a root server a **/var/nis/data/parent.object** to make this domain a part of the namespace above it. Only root servers can have parent objects. A parent objects describes the namespace above the NIS+ root. If this is an isolated domain, this flag should not be used. The argument to this flag tells the command what type of name server is serving the domain above the NIS+ domain. When clients attempt to resolve a name that is outside of the NIS+ namespace, this object is returned with the error `NIS_FOREIGNNS` indicating that a namespace boundary has been reached. It is up to the client to continue the name resolution process.

The parameter ″parent_domain″ is the name of the parent domain in a syntax that is native to that type of domain. The list of host names that follow the domain parameter are the names of hosts that serve the parent domain. It there is more than one server for a parent domain, the first host specified should be the master server for that domain.

Y Specifies that the parent directory is a NIS version 2 domain.

D Specifies that the parent directory is a DNS domain.

N *parent_domain_host***...**
Specifies that the parent directory is another NIS+ domain. This flag is useful for connecting a pre-existing NIS+ subtree into the global namespace.

-r Initializes the machine to be a NIS+ root server. This flag creates the file /var/nis/data/root.object and initializes it to contain information about this machine. It uses the sysinfo(2) system call to retrieve the name of the default domain.

Examples

1. To initialize a client, use:

```
nisinit -c -B
nisinit -c -H hostname
nisinit -c -C filename
```

2. To initialize a root master server, use:

```
nisinit -r
```

Initializing a Client

3. You can initialize a client in three different ways:
   - By host name
   - By broadcast
   - By cold-start file

Note: Each way has different prerequisites and associated tasks. For instance, before you can initialize a client by host name, the client's **/etc/hosts** file must list the host name you will use and **nsswitch.conf** file must have **files** as the first choice on the **hosts** line. Complete instructions for each method, including prerequisites and associated tasks, are provided in Initializing an NIS+ Client . Following is a summary of the steps that use the **nisinit** command.

4. To initialize a client by host name, use the **-c** and **-H** options, and include the name of the server from which the client will obtain its cold-start file:

   ```
   nisinit -c -H hostname
   ```

5. To initialize a client by cold-start file, use the **-c** and **-C** options, and provide the name of the cold-start file:

   ```
   nisinit -c -C filename
   ```

6. To initialize a client by broadcast, use the **-c** and **-B** options:

   ```
   nisinit -c -B
   ```

   Initializing the Root Master Server

7. To initialize the root master server, use the **nisinit -r** command:

   ```
   nisinit -r
   ```

Files

/var/nis/NIS_COLD_START	This file contains a list of servers, their transport addresses, and their Secure RPC public keys that serve the machine's default domain.
/var/nis/data/root.object	This file describes the root object of the NIS+ namespace. It is standard XDR-encoded NIS+ directory object that can be modified by authorized clients using the nis_modify() interface.
/var/nis/data/parent.object	This file describes the namespace that is logically above the NIS+ namespace. The most common type of parent object is a DNS object. This object contains contact information for a server of that domain.
/etc/hosts	Internet host table.

Related Information

The **nisclient** command, and **nisshowcache** command.

---

nisln Command

Purpose

Creates symbolic links between NIS+ objects and table entries.

Syntax

nisln [[**-L**] [**-D**] [*source*] [*target*]]

Description

The **nisln** command links objects to objects, or links objects to table entries. All NIS+ administration commands accept the **-L** flag, which directs them to follow links between NIS+ objects.

To create a link to another object or entry, you must have modify rights to the source object; that is, the one that will point to the other object or entry.

Notes:

1. A link cannot be created if it originates with a table entry.

2. Never link a cred table. Each **org_dir** directory should have its own cred table. Do not use a link to some other **org_dir** cred table.

Flags

-L	Follows link. If the **source** is itself a link, the new link will not be linked to it, but to that link's original source.
-D	Specifies a different set of defaults for the linked object. Defaults are described in Specifying Nondefault Security Values at Creation Time.

Example

To create a link between objects, specify both object names: first the *source*, and then the *target*. To create links between objects and entries use indexed names.

```
nisln source-object target-object
nisln [column=value,...],tablename target-object
```

---

nislog Command

Purpose

The **nislog** command displays the contents of the transaction log.

Syntax

nislog [**-h** *num* | **-t** *num*] [**-v**] [*directory*]...

Description

The **nislog** command displays the contents of the transaction log.

Each transaction consists of two parts: the particulars of the transaction and a copy of an object definition.

Here is an example that shows the transaction log entry that was made when the **wiz.com.** directory was first created. XID refers to the transaction ID.

```
rootmaster# /usr/sbin/nislog -h 1
NIS Log printing facility.
NIS Log dump:
        Log state : STABLE
        Number of updates    : 48
        Current XID          : 39
        Size of log in bytes : 18432
        ***UPDATES***@@@@@@@@@@@@@@@TRANSACTION@@@@@@@@@@@@@@@#00000,
        XID : 1
        Time        : Wed Nov 25 10:50:59 1992
 Directory   : wiz.com.
 Entry type   : ADD  Name
 Entry timestamp : Wed Nov 25 10:50:59 1992
 Principal       : rootmaster.wiz.com.
 Object name     : org_dir.wiz.com.
..................Object.....................
Object Name   : org_dir
Owner         : rootmaster.wiz.com.
Group         : admin.wiz.com.
Domain        : wiz.com.
Access Rights : r---rmcdr---r---
Time to Live  : 24:0:0
Object Type   : DIRECTORY
Name : `org_dir.wiz.com.'
Type: NIS
Master Server : rootmaster.wiz.com.
```

```
.
.
.........................................
@@@@@@@@@@@@@@@TRANSACTION@@@@@@@@@@@@@@@
#00000, XID : 2
```

Flags

-h *num* Display transactions starting with the head (beginning) of the log. If the number is omitted, the display begins with the first transaction. If the number 0 is entered, only the log header is displayed

-t *num* Display transactions starting backward from the end (tail) of the log. If the number is omitted, the display begins with the last transaction. If the number 0 is entered, only the log header is displayed

-v Verbose mode

---

nisls Command

Purpose

Lists the contents of an NIS+ directory.

Syntax

nisls [**-d**] [**-g**] [**-l**] [**-L**] [**-m**] [**-M**] [**-R**] [*Directory*...]

Description

The **nisls** command writes to standard output the contents of each directory specified in the parameter that is an NIS+ directory. If *Directory* specifies any other NIS+ object that is not a directory, **nisls** simply echoes the object's name. If no directory is given as a parameter, the first directory in the search path, the default, is listed (see **nisdefaults**).

Flags

-d Treats an NIS+ directory like other NIS+ objects instead of listing its contents.

-g Displays group owner instead of owner when using the **-l** flag to list in long format.

-l Lists in long format. The **-l** flag displays additional information about the *Directory* such as its type, creation time, owner, and permission rights.

-L Indicates that links are to be followed. If *Directory* actually points to a link, it is followed to a link object.

-m Displays modification time instead of creation time when using the **-l** flag to list contents in long format.

-M Specifies that the master server of the named directory returns the standard output of the **nisls** command. Using the **-M** flag guarantees that the most current information is listed.

-R Lists directories recursively. The **-R** flag displays the contents of each subdirectory contained in the directory specified in *Directory*.

Environment

NIS_PATH Searches each directory specified until the object is found if the NIS+ directory name is not fully qualified (see **nisdefaults**).

Exit Status

0 Successful completion.

1 An error occurred.

Examples

1. To list in short format the contents of **org.com.**, including its subdirectories, enter:
   ```
   nisls -R org.com.
   ```
2. To display detailed information about `rootmaster.org.com.`, including when it was last modified, enter:
   ```
   nisls -lm rootmaster.org.com.
   ```

Related Information

The **nisdefaults** command, **nisgrpadm** command, **nismatch** command, and **nistbladm** command.

---

nismatch Command

Purpose

Utility for searching NIS+ tables.

Syntax

nismatch [**-A**] [**-c**] [**-h**] [**-M**] [**-o**] [**-P**] [**-v**]

DESCRIPTION

The command **nisgrep** differs from the **nismatch** command in its ability to accept regular expressions for the search criteria rather than simple text matches.

Because **nisgrep** uses a callback function, it is not constrained to searching only those columns that are specifically made searchable at the time of table creation. This makes it more flexible, but slower, than **nismatch**.

In **nismatch**, the server does the searching; wheareas in **nisgrep**, the server returns all the readable entries and then the client does the pattern-matching.

In both commands, the parameter tablename is the NIS+ name of the table to be searched. If only one key or key pattern is specified without the column name, then it is applied searching the first column. Specific named columns can be searched by using the syntax.

When multiple columns are searched, only entries that match in all columns are returned. This is the equivalent of a logical join operation. **nismatch** accepts an additional form of search criteria, which is a NIS+ indexed name of the form:

Flags

-A Return the data within the table and all of the data in tables in the initial table's concatenation path.

-c Print only a count of the number of entries that matched the search criteria.

-h Display a header line before the matching entries that contains the names of the table's columns.

-M Master server only. Send the lookup to the master server of the named data. This guarantees that the most up to date information is seen at the possible expense that the master server may be busy.

-o Display the internal representation of the matching NIS+ object(s).

-P Follow concatenation path. Specify that the lookup should follow the concatenation path of a table if the initial search is unsuccessful.

-v Do not suppress the output of binary data when displaying matching entries. Without this option binary data is displayed as the string *\s-1BINARY\s0* .

1. **0** - Successfully matches some entries.
2. **1** - Successfully searches the table and no matches are found.
3. **2** - An error condition occurs. An error message is also printed.

Examples

1. This example searches a table named **passwd** in the **org_dir** subdirectory of the **zotz.com.domain**. It returns the entry that has the username of skippy.

   In this example, all the work is done on the server.

   ```
   nismatch\ name=skippy\ passwd.org_dir.zotz.com.
   ```

2. This example is similar to the one above except that it uses **nisgrep** to find all users in the table named **passwd** that are using either ksh (1) or csh (1).

   ```
   nisgrep\ 'shell=[ck]sh'\ passwd.org_dir.zotz.com.
   ```

3. NIS_PATH - If this variable is set, and the NIS+ table name is not fully qualified, each directory specified will be searched until the table is found (see **nisdefaults**, **niscat**, **nisls**, and **nistbladm**).

Related Information

The **nisgrep** command, **nisdefaults** command, **niscat** command, **nisls** command, and **nistbladm** command.

---

nismkdir Command

Purpose

Creates non-root NIS+ directories.

Syntax

nismkdir [**-D** *Defaults*] [**-m** *MasterHost* | **-s** *ReplicaHost*] *DirName*

Description

The **nismkdir** command creates subdirectories within an existing domain. It can also create replicated directories. Without any flags, the **nismkdir** command creates a subdirectory with the same master server and replica servers as its parent directory's. In addition, the **nismkdir** command can add a replica to an already existing directory.

A host that serves an NIS+ directory must be an NIS+ client in a directory above the one being served. The only exception is a root NIS+ server that acts as both client and server to the same NIS+ directory.

If the host's default domain is not the domain where the **nismkdir** command is executed, then the host name specified in the parameter with the **-s** or **-m** flags must be fully qualified.

> **Note:** You should use the **nisserver** command to create an NIS+ domain that consists of the named directory with the **org_dir** and **group_dir**.

Flags

-m *MasterHost*
: If the directory named by the *DirName* parameter does not yet exist, then the **-m** flag creates the new directory with *MasterHost* as its master server. If the directory named by *DirName* does exist, then the host named by the *MasterHost* parameter becomes its master server.

  > **Note:** To create a directory you must have create rights to the parent directory on that domain master server.

-s *ReplicaHost*

Adds a nonroot NIS+ directory and its master server to an existing system. Also, the **-s** flag can assign a new replica server to an existing directory. If *DirName* already exists, then the **nismkdir** command does not recreate it. Instead, it only assigns the new replica server to that existing directory.

After invoking the **-s** flag, you must run the **nisping** command from the master server on the directory that was added or assigned the replica server. You should include a **nisping** command for each directory in its master server's **cron** file so that it is pinged at least once every 24 hours before being updated.

Notes:

1. You cannot assign a server to support its parent domain, unless it belongs to the root domain.
2. Always run the **nismkdir** command on the master server. Never run **nismkdir** on the replica server. Running **nismkdir** on the replica server causes communication problems between the master and the replica.

-D *Defaults*

Specifies a different set of defaults for the new directory. The defaults string is a series of tokens each separated by a colon. These tokens represent the default values to be used for the generic object properties:

ttl=*Time*
: Sets the default time-to-live for objects created by the **nismkdir** command. The value *Time* is specified in the format defined by the **nischttl** command. The default value is 12h (12 hours).

owner=*Ownername*
: Specifies that the NIS+ principal *Ownername* should own the created object. The default for this value is the principal who is executing the command.

group=*Groupname*
: Specifies that the group *Groupname* should be the group owner for the object created. The default value is NULL.

access=*Rights*
: Specifies the set of access rights to be granted for the created object. The value *Rights* must be given in the format defined by the **nischmod** command. The default value is ——rmcdr—-r—-.

Environments

NIS_DEFAULTS Contains a defaults string that overrides the NIS+ standard defaults. If the **-D** flag is invoked then those values override both the **NIS_DEFAULTS** variable and the standard defaults.

NIS_PATH If the NIS+ directory name is not fully qualified, searches all directories specified until the directory is found (see **nisdefaults**).

Exit Status

This command returns the following the exit values:

0 Successful completion.
1 An error occurred.

Examples

1. To create the new directory `bar` under the `abc.com.` domain that shares the same master and replicas as the `abc.com.` directory, enter:

   ```
   nismkdir def.abc.com.
   ```

2. To create the new directory `def.abc.com.` that is not replicated under the `abc.com.` domain, enter:
   ```
   nismkdir\ \-m myhost.abc.com.\ def.abc.com.
   ```
3. To add a replica server of the `def.abc.com.` directory, enter:
   ```
   nismkdir\ \-s replica.abc.com.\ def.abc.com.
   ```

Files

Related Information

The **nischmod** command, **nisdefaults** command, **nisls** command, **nisrmdir** command, and **nisserver** command.

---

nismkuser Command

Purpose

Creates a new NIS+ user account.

Syntax

nismkuser [*Attribute*=*Value* ...] *Name*

Description

The **nismkuser** command creates a NIS+ user entry in the NIS+ domain. The *Name* parameter must be a unique 8-byte or less string. You cannot use the **ALL** or **default** keywords in the user name. By default, the **nismkuser** command creates a standard user account. To create an administrative user account, specify the **-a** flag.

> **Note:** You cannot use the **nismkuser** command to add users to an NIS+ groups. Use the **nisgrpadm** command to perform this function.

The **nismkuser** command will allow the input of the NIS+ user password at the time of user creation. If no password is given at user creation time, the NIS+ user's LOCAL and DES cred is created with the password `nisplus`. Later, passwords may be set or reset with the **passwd** command. New accounts are not disabled and are active after the **nismkuser** command completes.

> **Notes:**
> 1. Although this command allows the user to set the ″home″ directory for the NIS+ user, no actual physical directory is created if the directory does not already exist.
> 2. You need to have a group in *group.org_dir* with the gid that matches the new users gid first before you can add a user. The default gid for **nismkuser** is 1.

You can use the Web-based System Manager Users application or the System Management Interface Tool (SMIT) to run this command (under the NIS+ administration area).

Restrictions on Creating User Names

To prevent login inconsistencies, you should avoid composing user names entirely of uppercase alphabetic characters. While the **nismkuser** command supports multi-byte user names, it is recommended that you restrict user names to characters with the POSIX portable filename character set.

To ensure that your user database remains uncorrupted, you must be careful when naming users. User names must not begin with a - (dash), + (plus sign), @ (at sign), or ~ (tilde). You cannot use the keywords **ALL** or **default** in a user name. Additionally, do not use any of the following characters within a user-name string:

. Dot
: Colon
″ Double quote
Pound sign
, Comma
= Equal sign
\ Back slash
/ Slash
? Question mark
’ Single quote
` Back quote

Attention: You will not be allowed to create a NIS+ user with the identical name of a pre-existing NIS+ client or server name.

Finally, the *Name* parameter cannot contain any space, tab, or new-line characters.

Parameters

Attribute=*Value*	Initializes a user attribute. Refer to the **chuser** command for the valid attributes and values.
Name	Specifies a unique 8-byte or less string.

Valid Parameters

nismkuser will allow an administrator to enter the same attributes and parameters as you would with the **mkuser** command. However, only the following parameters will be used by the **nismkuser** command (the others will be ignored and not considered an error):
uid, **gid**, **gecos**, **shell**, **home**, **minage**,, **maxage**, **maxexpired**, **expires**.

Security

Access Control: This command should grant execute (x) access only to the root user and members of the security group. This command should be installed as a program in the trusted computing base (TCB). The command should be owned by the root user with the **setuid** (SUID) bit set.

Auditing Events:

Event	Information
USER_Create	user

Examples

1. To create the `davis` user account with the default values in the **/usr/lib/security/nismkuser.default** file, enter:

   ```
   nismkuser davis
   ```

2. To create the `davis` user account and set the **su** attribute to a value of `false`, enter:

   ```
   nismkuser su=false davis
   ```

Files

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

Related Information

The **chfn** command, **chgroup** command, **chgrpmem** command, **chsh** command, **chuser** command, **lsgroup** command, **lsuser** command, **mkgroup** 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 Standalone System Security in the *AIX 5L Version 5.2 Security Guide*.

For more information about administrative roles, refer to Users, Groups, and Passwords in the*AIX 5L Version 5.2 Security Guide*.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.2 Web-based System Manager Administration Guide*.

---

nisping Command

Purpose

Pings replica servers, telling them to ask the master server for updates immediately. When a replica responds, **nisping** updates the replica's entry in the root master server's niscachemgr cache file, **/var/nis/NIS_SHARED_DIRCACHE**.

Note: The replicas normally wait a couple of minutes before executing this request.

Syntax

To Display the Time of the Last Update

nisping [**-u** *domain*]

To Ping Replicas

nisping [**-H** *hostname*] [*domain*]

To Checkpoint a Directory

nisping [**-C** *hostname*] [*domain*]

Description

Before pinging, the command checks the time of the last update received by each replica. If it is the same as the last update sent by the master, it does not ping the replica.

The **nisping** command can also checkpoint a directory. This consists of telling each server in the directory, including the master, to update its information on disk from the domain's transaction log.

Flags

-u *domain*	Display the time of the last update; no servers are sent a ping.
-H *hostname*	Only the host **hostname** is sent the ping, checked for an update time, or checkpointed.
-C *hostname*	Send a request to checkpoint rather than a ping to each server. The servers schedule to commit all the transactions to stable storage.

Examples

Displaying the Time of the Last Update

Use the **-u** flag. It displays the update times for the master and replicas of the local domain, unless you specify a different domain name. It does not perform a ping.

/usr/lib/nis/nisping -u [domain**]**

Here is an example:

```
rootmaster# /usr/lib/nisping -u org_dir
 Last updates for directory wiz.com.:
 Master server is rootmaster.wiz.com.
        Last update occurred at Wed Nov 25 10:53:37 1992
 Replica server is rootreplica1.wiz.com.
        Last update seen was Wed Nov 25 10:53:37 1992
```

Pinging Replicas

You can ping all the replicas in a domain, or one in particular. To ping all the replicas, use the command without options:

/usr/lib/nis/nisping

To ping all the replicas in a domain other than the local domain, append a domain name:

/usr/lib/nis/nisping domainname

Here is an example that pings all the replicas of the local domain, **wiz.com.**:

```
rootmaster# /usr/lib/nis/nisping org_dir
 Pinging replicas serving directory wiz.com.:
 Master server is rootmaster.wiz.com.
        Last update occurred at Wed Nov 25 10:53:37 1992
 Replica server is rootreplica1.wiz.com.
        Last update seen was Wed Nov 18 11:24:32 1992

        Pinging ... rootreplica1.wiz.com.
```

Since the update times were different, it proceeds with the ping. If the times had been identical, it would not have sent a ping.

You can also ping all the tables in all the directories on a single specified host. To ping all the tables in all the directories of a particular host, us the **-a** flag:

/usr/lib/nis/nisping -a hostname

Checkpointing a Directory

To checkpoint a directory, use the **-C** flag:

/usr/lib/nis/nisping -C directory-name

All the servers that support a domain, including the master, transfer their information from their **.log** files to disk. This erases the log files and frees disk space. While a server is checkpointing, it will still answer requests for service, but it will be unavailable for updates.

Here is an example of **nisping** output:

```
rootmaster# /usr/lib/nis/nisping -C
Checkpointing replicas serving directory wiz.com. :
 Master server is rootmaster.wiz.com.
        Last update occurred at Wed May 25 10:53:37 1995
 Master server is rootmaster.wiz.com.
 checkpoint has been scheduled with rootmaster.wiz.com.
 Replica server is rootreplica1.wiz.com.
        Last update seen was Wed May 25 10:53:37 1995
 Replica server is rootreplica1.wiz.com.
 checkpoint has been scheduled with rootmaster.wiz.com.
```

nispopulate Command

Purpose

Populates the NIS+ tables in a NIS+ domain.

Syntax

nispopulate -Y [**-x**] [**-f**] [**-n**] [**-u**] [**-v**] [**-S 0** | **2**] [**-l** *network_passwd*] [**-d** *NIS+_domain*] **-h** *NIS_server_host* [**-a** *NIS_server_addr*] **-y** *NIS_domain* [*table*] ...

nispopulate -F [**-x**] [**-f**] [**-u**] [**-v**] [**-S 0** | **2**] [**-d** *NIS+_domain*] [**-l** *network_passwd*] [**-p** *directory_path*] [*table*] ...

nispopulate -C [**-x**] [**-f**] [**-v**] [**-d** *NIS+_domain*] [**-l** *network_passwd*] [*hosts* | *passwd*]

Description

The **nispopulate** command can be used to populate NIS+ tables in a specified domain from their corresponding files or NIS maps. The **nispopulate** command assumes that the tables have been created either through the **nisserver** command or the **nissetup** command.

The *table* argument accepts standard names and non-standard *key-value* type tables. See **nisaddent** for more information on *key-value* type tables. If the *table* argument is not specified, **nispopulate** will automatically populate each of the standard tables. These standard (default) tables are: **auto_master**, **auto_home**, **ethers**, **group**, **hosts**, **networks**, **passwd**, **protocols**, **services**, **rpc**, **netmasks**, **bootparams**, **netgroup**, **aliases**, and **shadow**.

> **Note:** The **shadow** table is only used when populating from files. The non-standard tables that **nispopulate** accepts are those of *key-value* type. These tables must first be created manually with the **nistbladm** command.

Use the first syntax (**-Y**) to populate NIS+ tables from NIS maps. The **nispopulate** command uses the **ypxfr** command to transfer the NIS maps from the NIS servers to the **/var/yp/***NIS_domain* directory on the local machine. Then, it uses these files as the input source.

> **Note:** *NIS_domain* is case sensitive. Make sure there is enough disk space for that directory.

Use the second syntax (**-F**) to populate NIS+ tables from local files. The **nispopulate** command will use those files that match the table name as input sources in the current working directory or in the specified directory.

When populating the **hosts** and **passwd** tables, the **nispopulate** command will automatically create the NIS+ credentials for all users and hosts that are defined in the **hosts** and **passwd** tables, respectively. A network password is required to create these credentials. This network password is used to encrypt the secret key for the new users and hosts. This password can be specified using the **-l** flag or it will use the default password, **nisplus**. This **nispopulate** will not overwrite any existing credential entries in the credential table. Use **nisclient** to overwrite the entries in the credential table. It creates both LOCAL and DES credentials for users and only DES credentials for hosts. To disable automatic credential creation, specify the **-S 0** flag.

The third syntax (**-C**) is used to populate NIS+ credential table with level 2 authentication (DES) from the passwd and hosts tables of the specified domain. The valid *table* arguments for this operation are **passwd** and **hosts**. If this argument is not specified, then it will use both **passwd** and **hosts** as the input source.

If **nispopulate** was earlier used with the **-S 0** flag, then no credentials were added for the hosts or the users. If later the site decides to add credentials for all users and hosts, then this (**-C**) flag can be used to add credentials.

The **nispopulate** command normally creates temporary files in the directory **/tmp**. You may specify another directory by setting the environment variable **TMPDIR** to your chosen directory. If **TMPDIR** is not a valid directory, then **nispopulate** will use **/tmp**.

Flags

-a *NIS_server_addr*	Specifies the IP address for the NIS server. This flag is only used with the **-Y** flag.
-C	Populates the NIS+ credential table from passwd and hosts tables using DES authentication (security level 2).
-d *NIS+_domain.*	Specifies the NIS+ domain. The default is the local domain.
-F	Populates NIS+ tables from files.
-f	Forces the script to populate the NIS+ tables without prompting for confirmation.
-h *NIS_server_host*	Specifies the NIS server hostname from where the NIS maps are copied from. This is only used with the **-Y** flag. This host must already exist in either the NIS+ **hosts** table or **/etc/hosts** file. If the hostname is not defined, the script will prompt you for its IP address, or you can use the **-a** flag to specify the address manually.
-l *network_passwd*	Specifies the network password for populating the NIS+ credential table. This is only used when you are populating the **hosts** and **passwd** tables. The default passwd is **nisplus**.
-n	Does not overwrite local NIS maps in **var/yp/***NISdomain* directory if they already exist. The default is to overwrite the existing NIS maps in the local **/var/yp/***NISdomain* directory. This is only used with the **-Y** flag.
-p *directory_path*	Specifies the directory where the files are stored. This is only used with the **-F** flag. The default is the current working directory.
-S 0 | **2**	Specifies the authentication level for the NIS+ clients. Level 0 is for unauthenticated clients, and no credentials will be created for users and hosts in the specified domain. Level 2 is for authenticated (DES) clients, and DES credentials will be created for users and hosts in the specified domain. The default is to set up with level 2 authentication (DES). There is no need to run the **nispopulate** command with the **-C** flag for level 0 authentication.
-u	Updates the NIS+ tables (that is, adds, deletes, modifies) from either files or NIS maps. This flag should be used to bring an NIS+ table up to date when there are only a small number of changes. The default is to add to the NIS+ tables without deleting any existing entries. Also, see the **-n** flag for updating NIS+ tables from existing maps in the **/var/yp** directory.
-v	Runs the script in verbose mode.
-x	Turns the ″echo″ mode on. The script just prints the commands that it would have executed. The commands are not actually executed. The default is off.
-Y	Populates the NIS+ tables from NIS maps.

-y *NIS_domain* Specifies the NIS domain to copy the NIS maps from. This is only used with the **-Y** flag. The default domainname is the same as the local domainname.

Examples

1. To populate all the NIS+ standard tables in the domain `xyz.ibm.com.` from NIS maps of the `yp.ibm.com` domain as input source where host `yp_host` is a YP server of `yp.ibm.com`, enter:

   ```
   /usr/lib/nis/nispopulate -Y -y yp.ibm.COM -h yp_host -d xyz.ibm.com.
   ```

2. To update all of the NIS+ standard tables from the same NIS domain and hosts shown above, enter:

   ```
   /usr/lib/nis/nispopulate -Y -u -y yp.ibm.COM -h yp_host -d xyz.ibm.com.
   ```

3. To populate the **hosts** table in domain `xyz.ibm.com.` from the hosts file in the **/var/nis/files** directory and using `somepasswd` as the network password for key encryption, enter:

   ```
   /usr/lib/nis/nispopulate -F -p /var/nis/files -l somepasswd hosts
   ```

4. To populate the passwd table in domain `xyz.ibm.com.` from the passwd file in the **/var/nis/files** directory without automatically creating the NIS+ credentials, enter:

   ```
   /usr/lib/nis/nispopulate -F -p /var/nis/files -d xys.ibm.com. -S 0 passwd
   ```

5. To populate the credential table in domain `xyz.ibm.com.` for all users defined in the passwd table, enter:

   ```
   /usr/lib/nis/nispopulate -C -d xys.ibm.com. passwd
   ```

6. To create and populate a non-standard key-value type NIS+ table, `private`, from the file **/var/nis/files/private:** (nispopulate assumes that the private.org_dirkey-value type table has already been created), enter:

   ```
   /usr/bin/nistbladm -D access=og=rmcd,nw=r \
        -c private key=S,nogw= value=,nogw= private.org.dir
   /usr/lib/nis/nispopulate -F -p /var/nis/files private
   ```

Files

/etc/hosts	Local host name database
/var/yp	NIS (YP) domain directory
/var/nis	NIS+ domain directory

Related Information

The **nistbladm** command, **nisaddcred** command, **nisaddent** command, **nisclient** command, **nisserver** command, **nissetup** command, **rpc.nisd** command, **ypxfr** command.

---

nisrm Command

Purpose

Removes NIS+ objects from the namespace.

Syntax

nisrm [**-i**] [**-f**] *Obj_name...*

Description

The **nisrm** command removes NIS+ objects from the NIS+ namespace. The **nisrm** command fails if the NIS+ master server is not running.

Notes: **nisrm** does not remove directories (see the **nisrmdir** command) nor non-empty tables (see the **nistbladm** command).

-i Sets the **nisrm** command in interactive mode. With the **-i** flag the **nisrm** command asks for confirmation before removing the specified object. If the object's name is not fully qualified then the **-i** flag is forced, preventing the unintended removal of another object.

-f Sets the **nisrm** command in force mode. If **nisrm** fails because you do not have the necessary permissions, **nischmod** is invoked and the removal is attempted again. If **nisrm** fails, it does not return an error message.

Examples

1. To remove the objects `xyz`, `abc`, and `def` from the namespace, enter:

```
nisrm xyz abc def
```

Environment

NIS_PATH With this variable set, if the NIS+ object name is not fully qualified, **nisrm** searches each directory indicated until the object is found.

Exit Status

0 Successful completion.

1 An error occurred.

Related Information

The **nischmod** command, **nisdefaults** command, **nisrmdir** command, **nistbladm** command, and the **rm** command.

---

nisrmdir Command

Purpose

Removes NIS+ objects from the namespace.

Syntax

nisrmdir [**-i**] [**-f**] [**-s** *Hostname*] *Dirname*

Description

The **nisrmdir** command removes existing NIS+ directories and subdirectories. The **nisrmdir** command can also remove replicas from serving a directory.

The **nisrmdir** command modifies the object that describes the directory (indicated in the parameter *Dirname*), then notifies each replica to remove it. If this notification fails, then the directory object is returned to its original state unless the **-f** flag is used.

nisrmdir fails if the NIS+ master server is not running.

-i Sets the **nisrmdir** command in interactive mode. With the **-i** flag, the **nisrm** command asks for confirmation before removing the specified object. If the directory's name in *Dirname* is not fully qualified, then the **-i** flag is forced, preventing the unintended removal of another directory.

-f Sets the **nisrm** command in force mode. The **-f** flag forces **nisrmdir** to succeed even though the command might not be able to contact the affected replica servers. Use this flag when you know that a replica is down and cannot respond to the removal notification. When the replica is finally rebooted, it reads the updated directory object, notes that it is no longer a replica for *Dirname*, and therefore, stops responding to lookups for that directory.

Note: You can clean up the files that held the removed directory by manually removing the appropriate files in the **/var/nis** directory.

-s *Hostname* Specifies that the server *Hostname* should be removed as a replica for the directory *Dirname*. If the **-s** flag is not used, then all replicas and the master server for *Dirname* are removed and the directory removed from the namespace.

Examples

1. To remove the directory `xyz` under the `abc.com.` domain, enter:
   ```
   nisrmdir xyz.abc.com.
   ```
2. To remove a replica serving the directory `xyz.abc.com.`, enter:
   ```
   nisrmdir -s replica.abc.com xyz.abc.com.
   ```
3. To force the removal of the directory `xyz.abc.com.` from the namespace, enter:
   ```
   nisrmdir -f xyz.abc.com.
   ```

Environment

NIS_PATH With this variable set, if the NIS+ directory name is not fully qualified, **nisrmdir** searches each directory indicated until the directory is found.

Exit Status

0 Successful completion.
1 An error occurred.

Related Information

The **nisdefaults** command and the **nisrm** command.

---

nisrmuser Command

Purpose

Removes a NIS+ user account.

Syntax

nisrmuser *Name*

Description

The **nisrmuser** command removes the NIS+ user account identified by the *Name* parameter. This command removes a user's attributes without removing the user's home directory and files. The user name must already exist as a string of 8 bytes or less.

Only the root user can remove administrative users. Administrative users are those users with **admin=true** set in the /**etc/security/user** file.

You can use the Web-based System Manager Users application or System Management Interface Tool (SMIT) to execute this command within the NIS+ administration section.

Security

Access Control: This command should grant execute (x) access only to the root user and members of the security group. This command should be installed as a program in the trusted computing base (TCB). The command should be owned by the root user with the **setuid** (SUID) bit set.

Auditing Events: ;

Event	Information
USER_Remove	user

Examples

1. To remove the user `davis` account and its attributes from the local system, enter:

   ```
   nisrmuser davis
   ```

Files

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

Related Information

The **chfn** command, **chgrpmem** command, **chsh** command, **chgroup** command, **chuser** command, **lsgroup** command, **lsuser** command, **mkgroup** command, **mkuser** command, **passwd** command, **pwdadm** command, **rmgroup** 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 Standalone System Security in the *AIX 5L Version 5.2 Security Guide*.

For more information about administrative roles, refer to Users, Groups, and Passwords in the*AIX 5L Version 5.2 Security Guide*.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.2 Web-based System Manager Administration Guide*.

---

nisserver Command

Purpose

Sets up NIS+ servers.

Syntax

To set up a root master server

/usr/lib/nis/nisserver -r [**-d** *Domain*] [**-f**] [**-g** *GroupName*] [**-l** *Password*] [**-v**] [**-x**] [**-Y**]

To set up a non-root master server

/usr/lib/nis/nisserver -M -d *Domain* [**-f**] [**-g** *GroupName*] [**-h** *HostName*] [**-v**] [**-x**] [**-Y**]

To set up a replica server

/usr/lib/nis/nisserver -R [**-d** *Domain*] [**-f**] [**-h** *HostName*] [**-v**] [**-x**] [**-Y**]

Description

The **nisserver** command is a shell script used to set up root master, non-root master, and replica NIS+ servers with level 2 security (DES).

When setting up a new domain, this script creates the NIS+ directories (including **groups_dir** and **org_dir**) and system table objects for the domain specified in *Domain*. However, **nisserver** does not populate tables with data. Use **nispopulate** to populate tables.

The **-r** flag is used to set up a root master server. In order to use this flag, you must be a superuser on the server where **nisserver** is executing. The **-M** flag is used to set up a non-root master server for the specified domain. To use this flag you must be an NIS+ principal on an NIS+ machine and have write permission to the parent directory of *Domain*. The new non-root master server must already be an NIS+ client (see the **nisclient** command) with the **rpc.nisd** daemon running. The **-R** flag is used to set up a replica server for both root and non-root domains. You must be an NIS+ principal on an NIS+ server and have write permission to the parent directory of the domain being replicated.

Flags

-d *Domain*	Specifies the NIS+ domain. The default is your local domain.
-f	Forces the NIS+ server setup without prompting for confirmation.
-g *GroupName*	Specifies the NIS+ group for the new domain. The **-g** flag is invalid with the **-R** flag. The default group is **admin**.
-h *HostName*	Specifies the host name for the NIS+ server. The server must be a valid host in the local domain. Use a fully qualified host name to specify a host outside of your local domain. The **-h** flag is only valid for setting up non-root master or replica servers. The default for the master server is to use the same list of servers as the parent domain's. The default for the replica server is to use the local host name.
-l *Password*	Specifies the network password for creating the credentials for the root master server. The **-l** flag is only valid with the **-r** flag. If you do not supply this flag, the **nisserver** script prompts you for the login password.
-M	Sets up the specified host as the master server. The **rpc.nisd** daemon must be running on that host before you execute the **nisserver** command with the **-M** flag.
-R	Sets up the specified host as the replica server. The **rpc.nisd** daemon must be running on that host before you execute the **nisserver** command with the **-M** flag.
-r	Sets up the server as the root master server.
-v	Runs the script in verbose mode.
-x	Turns the echo mode on.
-Y	Sets up an NIS+ server with NIS-compatibility mode. The default is no NIS-compatibility mode.

Examples

1. To set up a root master server for the domain `abc.com.`, enter:
   ```
   /usr/lib/nis/nisserver -r -d abc.com.
   ```
2. To set up a replica server for the domain `abc.com.` on the host `abcreplica`, enter:
   ```
   /usr/lib/nis/nisserver -R -d abc.com.
   /usr/lib/nis/nisserver -R -d abc.com. -h abcreplica
   ```
3. To set up a non-root master server for the domain `abc.xyz.com.` on the host `defhost` with the NIS+ group name as `admin-mgr.abc.xyz.com.` enter:
   ```
   /usr/lib/nis/nisserver -M -d abc.xyz.com.
   /usr/lib/nis/nisserver -M -d abc.xyz.com. -h defhost -g admin-mgr.abc.xyz.com.
   ```
4. To set up a non-root replica server for the domain `abc.xyz.com.` on `defhost`, enter:
   ```
   /usr/lib/nis/nisserver -R -d abc.xyz.com. -h defhost
   ```

Note: In each of the last three examples, the host must be an NIS+ client with the **rpc.nisd** daemon running before executing the command string.

Related Information

The **nisaddcred** command, the **nisclient** command, the **nisgrpadm** command, the **nisinit** command, the **nismkdir** command, the **nispopulate** command, the **nissetup** command, and the **rpc.nisd** daemon.

---

nissetup Command

Purpose

Initializes an NIS+ domain.

Syntax

/usr/lib/nis/nissetup [**-Y**] *NIS+Domain*

Description

The **nissetup** command initializes a domain to serve clients and to store system administration information. **nissetup** is a shell script that establishes an NIS+ domain to service clients needing to store system administration information in the domain *NIS+Domain*. That domain should already exist before executing **nissetup** (see **nismkdir** and **nisinit** for more information on how to create a domain).

An NIS+ domain consists of an NIS+ directory and its subdirectories, **org_dir** and **groups_dir**. The **org_dir** subdirectory stores system administration information and **groups_dir** stores information for group access control.

nissetup creates the subdirectories **org_dir** and **groups_dir** in *NIS+Domain*. Both **org_dir** and **groups_dir** are replicated on the parent domain's server. After the subdirectories are created, **nissetup** creates the default tables that NIS+ serves:

- **auto_master**
- **auto_home**
- **bootparams**
- **cred**
- **ethers**
- **group**
- **hosts**
- **mail_aliases**
- **netmasks**
- **networks**
- **passwd**
- **protocols**
- **rpc**
- **services** and
- **timezone**

The **nissetup** script uses the **nistbladm** command to create those tables. You can easily customize the script to add site-specific tables to be created at setup time.

Note: Although **nissetup** creates the default tables, it does not initialize them with data. Use the **nisaddent** command to accomplish this.

Normally, the **nissetup** command is executed only once per domain.

Flags

-Y Specifies that the domain is served as both an NIS+ and an NIS domain. The **-Y** flag makes all the system tables readable for unauthenticated clients; consequently, the domain is less secure.

Related Information

The **nisaddent** command, the **nisinit** command, the **nismkdir** command, and the **nistbladm** command.

---

nisshowcache Command

Purpose

Prints out the contents of the shared cache file.

Syntax

/usr/lib/nis/nisshowcache [**-v**]

Description

The **nisshowcache** command prints out the contents of the per-server NIS+ directory cache shared by all processes accessing NIS+ on the server. By default, **nisshowcache** only prints out the directory names in the cache along with the cache header. The shared cache is maintained by the **nis_cachemgr** command.

Flags

-v Sets the **nisshowcache** command in verbose mode. With the **-v** flag, **nisshowcache** prints out the contents of each directory object, including information on the server name and its universa addresses.

Files

/var/nis/NIS_SHARED_DIRCACHE contains the **nisshowcache** command.

Related Information

The **nis_cachemgr** command and the **syslogd** daemon.

---

nisstat Command

Purpose

Reports NIS+ server statistics.

Syntax

/usr/lib/nis/nisstat [**-H** *HostName*] [*DirName*]

Description

The **nisstat** command queries an NIS+ server for statistics about its operations. These statistics vary from release to release and between implementations. Not all statistics are available from all servers. If you request a statistic from a server that does not support it, **nisstat** simply returns **unknown statistic**.

By default, statistics are retrieved from the server(s) of the NIS+ directory for the default domain. If a directory is specified in *DirName*, then that directory's server is queried.

To retrieve a specific statistic, use one of these keywords:

root server Reports whether or not the server is a root server.

NIS compat mode Reports whether or not the server is running in NIS compat mode.

DNS forwarding in NIS mode Reports whether or not the server in NIS compat mode will forward host-lookup calls to DNS.

security level Reports the security level of the default server or the server specified in *HostName*.

serves directories Lists the directories served by the default server or the server specified in *HostName*.

Operations Returns results in the format

OP=*opname***:C=***calls***:E=***errors***:T=***micros*

opname
States the RPC procedure or operation.

calls States the number of calls to the RPC procedure made since the server began running.

errors States the number of errors that occurred while a call was being processed.

micros States the average amount of time (in microseconds) to complete the most recent 16 calls.

Directory Cache Reports the number of calls to the internal directory object cache, the number of hits on that cache, the number of misses, and the hit rate percentage.

Group Cache Reports the number of calls to the internal NIS+ group object cache, the number of hits on that cache, the number of misses, and the hit rate percentage.

Static Storage Reports the number of bytes the server allocated for its static storage buffers.

Dynamic Storage Reports the amount of heap the server process is currently using.

Uptime Reports the amount of time the service has been running.

Flags

-H *HostName* Indicates that only the server specified in *HostName* is queried by the **nisstat** command. By default, all servers for the directory are queried. If *HostName* does not serve the directory, no statistics are returned.

Environment

NIS_PATH If the NIS+: name is not fully qualified, searches each NIS+ directory specified until the directory is found.

Related Information

The **nisdefaults** command.

---

nistbladm Command

Purpose

Administers NIS+ tables.

Syntax

To add or overwrite table entries
nistbladm -a | **-A** [**-D** *Defaults*] { *Col_name=Value... Tbl_name* }

nistbladm -a | **-A** [**-D** *Defaults*] { *Entry_Name* }

Note: *Entry_Name* has the syntax `[column=value],table`.

To create an NIS+ table
nistbladm -c [**-D** *Defaults*] [**-p** *Path*] [**-s** *Sep*] *Type Col_name*=[**S**] [**I**] [**C**] [**B**] [**X**] [*Access*]... *Tbl_name*

Note: The flags after *Col_name* must be comma separated.

Example:

```
nistbladm -c hobby_tbl name=S,a+r,o+m hobby=S,a+r hobbies.abc.com.
```

To delete an entire NIS+ table
nistbladm -d *Tbl_name*

To edit table entries
nistbladm -m | **-E** *Col_name=Value... Entry_name*

To remove table entries
nistbladm -r | **-R** { [*Col_name=Value...*] *Tbl_name* }

nistbladm -r | **-R** { *Entry_name* }

To update a table's attributes
nistbladm -u [**-p** *Path*] [**-s** *Sep*] [**-t** *Type*] [*Col_name=Access...*] *Tbl_name*

Description
The **nistbladm** command is used to administer NIS+ tables. It performs five primary operations: creating tables, deleting tables, adding table entries, modifying table entries, and removing table entries.

Though NIS+ does not restrict the size of tables or entries, the size of data affects the performance and the disk space requirements of the NIS+ server. NIS+ is not designed to store huge amounts of data, such as files. Instead, store pointers to files located on other servers. NIS+ can support up to 10,000 objects totaling 10M bytes. If the you need more storage space, create the domain hierarchy, or use the data stored in the tables as pointers to the actual data, instead of storing the actual data in NIS+.

To create a table, its directory must already exist and you must have create rights to that directory. You must specify a table name, table type, and a list of column definitions. *Type* is a string that acts as a standard by which NIS+ verifies that entries are of the correct type.

To delete a table, you must have destroy rights to the directory where it is stored. To modify entries, whether adding, changing, or deleting, you must have modify rights to the tables or individual entries.

Flags

-a Adds a new entry to an NIS+ table. Create the entry's contents by supplying *Col_name=Value* pairs on the command line.

Notes:

1. You must specify a value for each column when adding an entry to an NIS+ table.
2. When entering the value string, enclose terminal characters in single quotation marks (') or double quotation marks (″). Those characters are the equals sign (=), comma (,), left bracket ([), right bracket (]), and space ( ). They are sparsed by NIS+ within an indexed name.

With the **-a** flag, the **nistbladm** command reports an error if you attempt to add an entry that would overwrite a pre-existing value in the desired column. The **nistbladm** command does not automatically overwrite pre-existing entry values. (See the **-A** flag for information about overwriting entries.)

-A Forces the **nistbladm** command to overwrite a pre-existing entry value. Even if *Col_name* already contains a value, **nistbladm** overwrites the old value with the new value. Unlike with the **-a** flag, the **nistbladm** command does not return an error.

-c *Tbl_name* Creates a new NIS+ table named in the parameter *Tbl_name*. When creating a table, you must specify a table type, entry type, and a list of column definitions. The syntax for column definitions is *Col_name*=[*Flags*] [*Access*]. The parameter *Flags* can have these possible values:

S Specifies that searches can be performed on the column's values.

I Specifies that searches ignore the case of column values. This flag is only valid in combination with the **S** flag.

C Encrypts the column's values.

B Sets the column's values as binary data. If the **B** flag is not set, column values are null-terminated ASCII strings. This flag is only valid in combination with the **S** flag.

X Sets the column's values as XDR-encoded data. The **X** flag is only valid in combination with the **B** flag.

The newly created table must contain at least one column in number and at least one searchable column; in other words, if *Tbl_name* only has only one column, that column must be searchable.

-d *Tbl_name* Deletes the entire table indicated in the parameter *Tbl_name*. The table must be empty before you delete it. (Use the **-R** flag to delete a table's contents.)

-D Specifies a set of defaults to be used when new objects are created. The defaults string is a series of tokens separated by colons. These tokens represent the default values to be used for the generic object properties.

ttl=*Time*
Sets the default time-to-live for objects created by the **nistbladm** command. The value *Time* must be given in the format defined by the **nischttl** command. The default value is 12 hours.

owner=*Ownername*
Specifies that the NIS+ principal *Ownername* should own the created object. The default value is the the same as the principal who executes the **nistbladm** command to create the object.

group=*Groupname*
Specifies that the group *Groupname* should be the group owner for the object created. The default value is NULL.

access=*Rights*
Specifies the set of access rights to be granted for the given object. The value *Rights* must be given in the format defined by the **nischmod** command. The default value is ——rmcdr—-r—-.

-e *Entry_name* Edits the entry specified by *Entry_name*. *Entry_name* must uniquely identify only one single entry. While editing the value of *Entry_name*, you can also change that entry's indexed name.
Note: If the entry's new indexed name (resulting from the edit) matches that of another's entry, the **nistbladm** command fails and returns an error message.

-E *Entry_name* Edits the entry specified by *Entry_name*. *Entry_name* must uniquely identify only one single entry.
Note: If the new indexed name matches that of another entry, then the **-E** flag automatically overwrites that existing entry with the entry just edited. So, in effect, two entries are being replaced by one.

-m Same functionality as **-E**.

-r Removes an entry from a table. Either identify the entry by its indexed name in *Entry_value*, or by a series of *Col_name*=*Value* pairs on the command line. With the **-r** flag, the **nistbladm** command fails when the indexed name or the column=value pairs match more than one entry.

-R Removes multiple entries from a table. The **-R** flag forces the **nistbladm** command to remove all entries that match the criterion for removal. If that criterion is null—if you do not specify column=value pairs or an indexed name—then **all** entries from the table are removed.

-u Updates attributes of a table. This allows the concatenation path, separation character, column access rights, and table type string of a table to be changed. Neither the number of columns nor the number of searchable columns can be changed with this flag.

-p *Path* Specifies the table's search path when creating or modifying a table. When you invoke the **nis_list** function, you can specify the flag **FOLLOW_PATH** to tell the client library to continue searching tables in *Path* if the search criteria does not yield any entries. The path consists of an ordered list of table names separated by colons. The names in the path must be fully qualified.

-s *Sep* Specifies the table's separator character when creating or modifying a table. The separator character is used by the **niscat** command when writing tables to standard output. The purpose of the separator character is to separate column data when the table is in ASCII form. The default value is a <space>.

-t *Type* Specifies the tables's *Type* string when modifying a table.

Exit Status

0 Successful completion.
1 An error occurred.

Environment Variables

NIS_DEFAULTS Contains a defaults string that overrides the NIS+ standard defaults. However, if you specify different values with the **-D** flag, then those values overrides both the **NIS_DEFAULTS** variable and the standard defaults.

NIS_PATH If *Tbl_name* is not fully qualified, then setting this variable instructs **nistbladm** to search each directory specified until the table is found.

Examples

1. To create a table named `hobbies` in the directory `abc.com.` of the type `hobby_tbl` with two searchable columns `name` and `hobby`, type:

   ```
   nistbladm -c hobby_tbl name=S,a+r,o+m hobby=S,a+r hobbies.abc.com.
   ```

   The column `name` has read access for all (owner, group, and world) and modify access for only the owner. The column `hobby` has read access for all but cannot be modified by anyone.

If access rights are not specified, then the table access rights would be either the standard defaults or those specified by the **NIS_DEFAULTS** variable.

2. Too add entries to the `hobbies` table, type:

```
nistbladm -a name=bob hobby=skiing hobbies.abc.com.

nistbladm -a name=sue hobby=skiing hobbies.abc.com.

nistbladm -a name=ted hobby=swimming hobbies.abc.com.
```

3. To add the concatenation path, type:

```
nistbladm -u -p hobbies.xyz.com.:hobbies.def.com. hobbies
```

4. To delete skiing-enthusiasts from the table, type:

```
nistbladm -R hobby=skiing hobbies.abc.com.
```

   Note: Using the **-r** flag in this example would fail because two entries contain the value `skiing`.

5. To create a table with a column that is named with no flags set, type:

```
nistbladm -c notes_tbl_ name=S,a+r,o+m note=notes.abc.com.
```

   This command string creates the table `notes.abc.com.` of the type `notes_tbl` with the two columns, `name` and `note`. The `note` column is not searchable.

Related Information

The **niscat** command, the **nischmod** command, the **nischown** command, the **nisdefaults** command, the **nismatch** command, and the **nissetup** command.

---

nistest Command

Purpose

Returns the state of the NIS+ namespace using a conditional expression.

Syntax

nistest [[**-A**] [**-L**] [**-M**] [**-P**]] [**-a** | **-t** *Type*] *Object*

nistest [**-A**] [**-L**] [**-M**] [**-P**] [**-a** *Rights*] *IndexedName*

Description

The **nistest** command provides a way for shell scripts and other programs to test for the existence, type, and access rights of objects and entries. Entries are named using indexed names (see the **nismatch** command.)

Flags

-A	Specifies that all of the data within the table and all of the data in tables in the initial table's concatenation path be returned. This flag is only valid when using indexed names or following links.
-L	Follow links. If the object named by *Object* or the tablename component of *IndexedName* names a LINK type object, the link is followed when this switch is present.
-M	Specifies that the lookup should only be sent to the master server of the named data. This guarantees that the most up to date information is seen at the possible expense that the master server may be busy.
-P	Specifies that the lookup should follow the concatenation path of a table if the initial search is unsuccessful. This flag is only valid when using indexed names or following links.

-a *Rights* Verifies that the current process has the desired or required access rights on the named object or entries. The access rights are specified in the same way as the **nischmod** command.

-t *Type* Tests the type of *Object*. The value of *type* can be one of the following:

G Return true if the object is a group object.

D Return true if the object is a directory object.

T Return true if the object is a table object.

L Return true if the object is a link object.

P Return true if the object is a private object.

RETURN VALUES

0	Success.
1	Failure due to object not present, not of specified type and/or no such access.
2	Failure due to illegal usage.

Examples

1. When testing for access rights, **nistest** returns success (0) if the specified rights are granted to the current user. Thus testing for access rights

   ```
   nistest \-a w=mr skippy.domain
   ```

   Tests that all authenticated NIS+ clients have read and modify access to the object named `skippy.domain`.
2. Testing for access on a particular entry in a table can be accomplished using the indexed name syntax. The following example tests to see if an entry in the password table can be modified.

   ```
   nistest \-a o=m '[uid=99],passwd.org_dir'
   ```

Environment

NIS_PATH

If this variable is set, and the NIS+ name is not fully qualified, each directory specified will be searched until the object is found (see **nisdefaults**).

Related Information

The **nischmod** command and **nisdefaults** command.

---

nistoldif Command

Purpose

Migrate user, group, name resolution, and rpc data to rfc 2307-compliant form.

Syntax

nistoldif -d *Suffix* [**-a** *BindDN* **-h** *Host* **-p** *Password* [**-n** *Port*]] [**-f** *Directory*] [**-y** *domain*] [-S *Schema*] [**-k** *KeyPath* **-w** *SSLPassword*] [**-s** *Maps*]

Description

The **nistoldif** command converts the data from **passwd**, **group**, **hosts**, **services**, **protocols**, **rpc**, **networks**, and **netgroup** into forms compliant with rfc2307. It will first attempt to read data from NIS, and if it cannot find a NIS map it will fall back to the flat files.

If the server information (the **-a**, **-h**, and **-p** flags) is given on the command line, data will be written directly to the server. If any data conflicts with an entry already on the server, either because the entry already exists, or because the **uid** or **gid** already exists, a warning will be printed. If the server information is not given, the data will be written to **stdout** in LDIF. In either case, **nistoldif** does not add an entry for the suffix itself; if that entry does not exist, attempts to add data to the server will fail. This entry will be added during server setup, usually by the **mksecldap** command.

Translation is not exact. Because of the limitations of the rfc2307 definitions, some attributes are defined in a case-insensitive way; for example, TCP, Tcp, and tcp are all the same protcol name to the LDAP server. Uids and gids greater than 2^31-1 will be translated to their negative twos complement equivalent for storage.

Flags

-a Specifies the administrative bind DN used to connect to the LDAP server. If this flag is used, **-h** and **-p** must also be used, and data will be written directly to the LDAP server.

-d Specifies the suffix that the data should be added under.

-f Specifies the directory to look for flat files in. If this flag is not used, **nistoldif** will look for flat files in **/etc**.

-h Specifies the host name which is running the LDAP server. If this flag is used, **-a** and **-p** must also be used, and data will be written directly to the LDAP server.

-k Specifies the SSL key path. If this flag is used, **-w** must also be used.

-n Specifies the port to connect to the LDAP server on. If this flag is used, **-a**, **-h** and **-p** must also be used; if it is not used, the default LDAP port is used.

-p Specifies the password used to connect to the LDAP server. If this flag is used, **-a** and **-h** must also be used, and data will be written directly to the LDAP server.

-s Specifies a set of maps to be written to the server. This flag should be followed by a list of letters representing the maps that should be migrated. If this flag is not used, all maps will be migrated. The letters are: **e** for netgroup, **g** for group, **h** for hosts, **n** for networks, **p** for protocols, **r** for rpc, **s** for services, and **u** for passwd.

-S Specifies the LDAP schema to use for users and groups. This can be either RFC2307 or RFC2307AIX; RFC2307AIX gives extended AIX schema support. If this flag is not used, RFC2307 is the default.

-w Specifies the SSL password. If this flag is used, **-k** must also be used.

-y Specifies the NIS domain to read maps from. If this flag is not used, the default domain will be used.

Exit Status

This command returns the following exit values:

0 No errors occured. Note that failure to find a map is not considered an error.

>0 An error occurred.

Security

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

Examples

1. To migrate the NIS maps from the domain **austin.ibm.com** (falling back to the flat files in **/tmp/etc**) to LDIF under the suffix **cn=aixdata**, type:

   ```
   nistoldif -d cn=aixdata -y austin.ibm.com -f /tmp/etc > ldif.out
   ```

2. To migrate the hosts and services maps from the default domain (falling back to the flat files in **/etc**) to the LDAP server **ldap.austin.ibm.com** with administrator bind DN**cn=root** and password `secret` under the suffix **cn=aixdata**, type:

   ```
   nistoldif -d cn=aixdata -h ldap.austin.ibm.com -a cn=root -p secret -s hs
   ```

Files

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

Related Information

The **mksecldap** command.

---

nisupdkeys Command

Purpose

Updates the public keys in NIS directory objects.

Syntax

/usr/lib/nis/nisupdkeys [**-a**] | [**-C**] [**-H** *Hostname*] [**-s**] [*Dirname*]

Description

The **nisupdkeys** command updates the public keys in an NIS+ directory object. When the public key for an NIS+ server is changed, the new key must be propagated to all directory objects that reference that server. **nisupdkeys** reads a directory object and attempts to copy the public key for each server of that directory. The key is then placed in the directory object and then the object is modified to reflect the new key.

If *Dirname* exists, then its directory object is updated. If not, then the directory object for the default domain is updated. **nisupdkeys -s** obtains a list of all the directories served by *Hostname* and updates those directory objects, assuming that the caller has the necessary permission rights. That list of directories can also be obtained by the **nisstat** command.

Before you run **nisupdkeys**, make sure you have propagated the new address/public key to all replica servers.

Flags

-a	Updates the universal addresses of the NIS+ servers in the directory object. The **-a** flag only works for the TCP/IP family of transports. You should use this flag when the IP address of the server is changed. The new address is resolved using **gethostname** on that server. In order for this resolution to work, the **/etc/nsswitch.conf** file must point to the correct source of the server's entry.
-C	Clears the public key. Communication with a server that has no public key does not require the use of a secure remote procedure call.
-H *Hostname*	Updates the keys of the server named *Hostname* for the current domain directory object. If the host name is not fully qualified, then **nisupdkeys** assumes the server is in the default domain. If *Hostname* does not serve the directory, then nothing happens.
-s	Updates all the NIS+ directory objects served by the server *Hostname*, assuming that you have the necessary permission rights. If you do not have permission to update the directory objects, those updates fail and you will be notified. If the **rpc.nisd** on *Hostname* can't return the list of servers it serves, **nisupdkeys** returns an error message. Then you must invoke the **nisupdkeys** multiple times, once per NIS+ directory the **rpc.nisd** serves.
Dirname	Updates the keys of the directory object for the directory *Dirname*.

Examples

1. To update the keys for servers of the `abc.def.` domain, enter:

   ```
   nisupdkeys abc.def.
   ```

2. To update the keys for host `xyzserver` that serves the `abc.def.` domain, enter:
   ```
   nisupdkeys -H xyzserver abc.def.
   ```
3. To clear the keys for host `xyzserver` in the `abc.def.` domain, enter:
   ```
   nisupdkeys -CH xyzserver abc.def.
   ```
4. To update the keys in all directory objects served by `xyzserver`, enter:
   ```
   nisupdkeys -sH xyzserver
   ```

Security

Access Control: To use the **nisupdkeys** command, you must have modify rights to the NIS+ directory object.

Files

/usr/lib/nis	Directory where the **nisupdkeys** command resides.

Related Information

The **chkey** command, the **nisaddcred** command, and the **niscat** command.

The **gethostbyname** subroutine.

---

nl Command

Purpose

Numbers lines in a file.

Syntax

nl [**-b** *Type*] [**-f** *Type*] [**-h** *Type*] [**-l** *Number*] [**-d** *Delimiter*] [**-i** *Number*] [**-n** *Format*] [**-v** *Number*] [**-w** *Number*] [**-p**] [**-s** *Separator*] [*File*]

Description

The **nl** command reads the *File* parameter (standard input by default), numbers the lines in the input, and writes the numbered lines to standard output. In the output, the **nl** command numbers the lines on the left according to the flags you specify on the command line.

The input text must be written in logical pages. Each logical page has a header, a body, and a footer section (you can have empty sections). Unless you use the **-p** flag, the **nl** command resets the line numbers at the start of each logical page. You can set line-numbering flags independently for the header, body, and footer sections (for example, the header and footer lines can be numbered while the text lines are not).

Signal the start of logical-page sections with lines in the file that contain only the following delimiter characters:

Line Contents	Start Of
\:\:\:	Header
\:\:	Body
\:	Footer

You can name only one file on the command line. You can list the flags and the file name in any order.

Flags

All the parameters are set by default. Use the following flags to change these default settings. Except for the **-s** flag, enter a **-n** flag without a variable to see its default value.

-b *Type* Chooses which body section lines to number. Recognized values for the *Type* variable are:

- **a** Numbers all lines
- **t** Does not number lines that are blank or lines that contain any non-graphic character such as a tab within them. (default)
- **n** Does not number any lines
- **p***Pattern* Numbers only those lines specified by the *Pattern* variable.

-d *Delimiter* Uses the two characters specified by the *Delimiter* variable as the delimiters for the start of a logical page section. The default characters are \: (backslash, colon). You may specify two ASCII characters, two 1-byte extended characters, or one extended character. If you enter only one 1-byte character after the **-d** flag, the second character remains the default (a colon). If you want to use a backslash as a delimiter, enter two backslashes (\\).

-f *Type* Chooses which logical-page footer lines to number. The possible values for the *Type* variable are the same as the **-b** flag. The default value of the *Type* variable is **n** (no lines numbered).

-h *Type* Chooses which logical-page header lines to number. The possible values for the *Type* variable are the same as the **-b** flag. The default value of the *Type* variables **n** (no lines numbered).

-i *Number* Increments logical-page line numbers by the number specified in the *Number* variable. The default value of the *Number* variable is 1. The range of the *Number* variable is from 1 to 250.

-l *Number* (Lowercase L) Uses the value specified in the *Number* parameter as the number of blank lines to count as one. For example, `-l3` numbers every third blank line in a series. The default value of the *Number* variable is 1. This flag works when the **-ha**, **-ba**, or **-fa** option is set. The range of the *Number* variable is from 1 to 250.

-n *Format* Uses the value of the *Format* variable as the line numbering format. Recognized formats are:

- **ln** Left-justified, leading zeros suppressed
- **rn** Right-justified, leading zeros suppressed (default)
- **rz** Right-justified, leading zeros kept

-p Does not restart numbering at logical page delimiters.

-s *Separator* Separates the text from its line number with the character specified in the *Separator* variable. The default value of the *Separator* variable is a tab character.

-v *Number* Sets the initial logical-page line number to the value specified by the *Number* variable. The default value of the *Number* variable is 1. The range of the *Number* variable is from 0 to 32767.

-w *Number* Uses the value specified by the *Number* variable as the number of characters in the line number. The default value of the *Number* variable is 6. The range of the *Number* variable is from 1 to 20.

Exit Status

This command returns the following exit values:

0 Successful completion.
>0 An error occurred.

Examples

1. To number only the non-blank lines, enter:

```
nl  chap1
```

This displays a numbered listing of `chap1`, numbering only the non-blank lines in the body sections. If `chap1` contains no \:\:\+:, or \: delimiters, then the entire file is considered the body.

2. To number all lines:

```
nl  -ba  chap1
```

This numbers all the lines in the body sections, including blank lines. This form of the **nl** command is adequate for most uses.

3. To specify a different line number format, enter:

```
nl  -i10  -nrz  -s::  -v10  -w4  chap1
```

This numbers the lines of `chap1` starting with ten (`-v10`) and counting by tens (`-i10`). It displays four digits for each number (`-w4`), including leading zeros (`-nrz`). The line numbers are separated from the text by two colons (`-s : :`).

For example, if `chap1` contains the text:

```
A  not-so-important
note  to  remember:

You  can't  kill  time
without  injuring  eternity.
```

then the numbered listing is:

```
0010::A  not-so-important
0020::note  to  remember

0030::You  can't  kill  time
0040::without  injuring  eternity.
```

Note that the blank line was not numbered. To do this, use the **-ba** flag as shown in example 2.

Files

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

Related Information

Files and Input and Output Redirection in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices*.

The **pr** command.

---

nlssrc Command

Purpose

Gets the status of a subsystem or a group of subsystems in canonical form.

Syntax

nlssrc [**–h** *host*] **–a**

nlssrc [**–h** *host*] **–g** *group_name*

nlssrc [**–h** *host*] [**–l**] [**–c**] **–s** *subsystem_name*

nlssrc [**–h** *host*] [**–l**] [**–c**] **–p** *subsystem_pid*

The syntax for the first two usages of **nlssrc** will generate the exact same output as **lssrc**. The syntax for the last two usages will generate the output in the canonical form as **lssrc**.

Description

Use the **nlssrc** command to get the status of a subsystem or a group of subsystems in canonical form. Use the **nlssrc -c** command to get language-independent output for supported subsystems from the **lssrc** command. The status is displayed in English regardless of the installed language locale. If the **–c** flag is not present, the **nlssrc** command will invoke the **lssrc** command that uses the daemon's locale.

Flags

–a Lists the current status of all defined subsystems.

–c Requests the canonical **lssrc** output of the supported subsystems.

–g *group_name*
Specifies a group of subsystems to get status for. The command is unsuccessful if the *group_name* parameter is not contained in the subsystem object class.

–h *host*
Specifies the foreign host on which this status action is requested. The local user must be running as root. The remote system must be configured to accept remote System Resource Controller (SRC) requests. That is, the **srcmstr** daemon (see **/etc/inittab**) must be started with the **–r** flag and the **/etc/hosts.equiv** file or the **.rhosts** file must be configured to allow remote requests.

–l Requests that a subsystem send its current status in long form. Long status requires that a status request be sent to the subsystem; it is the responsibility of the subsystem to return the status.

–p *subsystem_pid*
Specifies a particular instance of the *subsystem_pid* parameter to get status for, or a particular instance of the subsystem to which the status subserver request is to be taken.

–s *subsystem_name*
Specifies a subsystem to get status for. The *subsystem_name* parameter can be the actual subsystem name or the synonym name for the subsystem. The command is unsuccessful if the *subsystem_name* parameter is not contained in the subsystem object class.

Security

You do *not* need **root** authority to run this command.

Exit Status

0 Command has run successfully.

1 Command was not successful.

Restrictions

This command applies to the **cthags** and **cthats** subsystems only.

Standard Output

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

Standard Error

Error messages are written to standard error (and to the **ctsnap.***host_name***.***nnnnnnnn***.log** file).

Examples

1. To get **nlssrc** output in English from a subsystem called **ctsubsys**, enter:

```
nlssrc -c -ls ctsubsys
```

2. The following example shows the same information in different formats:

```
nlssrc -ls ctsubsys (locale-dependent)

Subsystem  Group     PID   Status
ctsubsys   ctsubsys  6334  active
2 locally-connected clients.  Their PIDs:
15614 23248
HA Subsystem domain information:
Domain established by node 5
Number of groups known locally: 1
                    Number of          Number of local
Group Name          providers          providers/subscribers
ha_filesys             7                   1        0
```

```
nlssrc -ls ctsubsys -c (canonical form)

Number of local clients: 2
PIDs: 15614 23248
HA Subsystem domain information:
Domain established by node 5.
Number of known local groups: 1
Group Name: ha_filesys
     Providers: 7
     Local Providers: 1
     Local Subscribers: 0
```

Location

/usr/sbin/rsct/bin/nlssrc Contains the **nlssrc** command

Files

/tmp/ctsupt Location of the default directory that contains the output files.

/tmp/ctsupt/ctsnap.*host_name***.***nnnnnnnn***.log**
Location of the log file of the command execution, where *nnnnnnnn* is a timestamp and *host_name* is the name of the host on which the command is running.

tmp/ctsupt/ctsnap.*host_name***.***nnnnnnnn***.tar.Z**
Location of the compressed tar file that contains the collected data, where *nnnnnnnn* is a timestamp and *host_name* is the name of the host on which the command is running.

Related Information

Commands: **lssrc**(1)

---

nm Command

Purpose

Displays information about symbols in object files, executable files, and object-file libraries.

Syntax

nm [**-A**] [**-C**] [**-X** {**32**|**64**|**32_64**}] [**-f**] [**-h**] [**-l**] [**-p**] [**-r**] [**-T**] [**-v**] [**-B** | **-P**] [**-e** | **-g** | **-u**] [**-d** | **-o** | **-x** | **-t** *Format*] *File* ...

Description

The **nm** command displays information about symbols in the specified *File*, which can be an object file, an executable file, or an object-file library. If the file contains no symbol information, the **nm** command reports the fact, but does not interpret it as an error condition. The **nm** command reports numerical values in decimal notation by default.

The **nm** command writes the following symbol information to standard output:

- **Library** or **Object Name**

  The **nm** command reports either the library or the object name associated with the file only if you specify the **-A** option.
- **Symbol Name**
- **Symbol Type**

  The **nm** command represents the file's symbol type with one of the following characters (with weak characters treated as global):

A Global absolute symbol.
a Local absolute symbol.
B Global bss symbol.
b Local bss symbol.
D Global data symbol.
d Local data symbol.
f Source file name symbol.
T Global text symbol.
t Local text symbol.
U Undefined symbol.

- **Value**
- **Size**

  The **nm** command reports the size associated with the symbol, if applicable.

Flags

-A Displays either the full path name or library name of an object on each line.

-B Displays output in the Berkeley Software Distribution (BSD) format:

```
value   type   name
```

-C Suppresses the demangling of C++ names. The default is to demangle all C++ symbol names.
Note: Symbols from C++ object files have their names demangled before they are used.

-d Displays a symbol's value and size as a decimal. This is the default.

-e Displays only static and external (global) symbols.

-f Displays full output, including redundant .text, .data, and .bss symbols, which are normally suppressed.

-g Displays only external (global) symbols.

-h Suppresses the display of output header data.

-l Distinguishes between WEAK and GLOBAL symbols by appending a * to the key letter for WEAK symbols. If used with the **-P** option, the symbol type for weak symbols is represented as follows:

V Weak Data Symbol

W Weak Text Symbol

w Weak Undefined Symbol

Z Weak bss Symbol

-o Displays a symbol's value and size as an octal rather than a decimal number.

-P Displays information in a standard portable output format:

```
library/object name   name   type   value   size
```

This format displays numerical values in hexadecimal notation, unless you specify a different format with the **-t**, **-d**, or **-o** flags.

The **-P** flag displays the **library/object name** field only if you specify the **-A** flag. Also, the **-P** flag displays the **size** field only for symbols for which size is applicable.

-p Does not sort. The Ouput is printed in symbol-table order.

-r Sorts in reverse order.

-T Truncates every name that would otherwise overflow its column, making the last character displayed in the name an asterisk. By default, **nm** displays the entire name of the symbols listed, and a name that is longer than the width of the column set aside for it causes every column after the name to be misaligned.

-t *Format* Displays numerical values in the specified format, where the *Format* parameter is one of the following notations:

d Decimal notation. This is the default format for the **nm** command.

o Octal notation.

x Hexadecimal notation.

-u Displays only undefined symbols.

-v Sorts output by value instead of alphabetically.

-x Displays a symbol's value and size as a hexadecimal rather than a decimal number.

-X *mode* Specifies the type of object file **nm** 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

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 **nm** to process any 64-bit objects and ignore 32-bit objects. The **-X** flag overrides the **OBJECT_MODE** variable.

Note: The **nm** command supports the **—** (double hyphen) flag. This flag distinguishes a *File* operand if the file name can be misinterpreted as an option. For example, to specify a file name that begins with a hyphen, use the **—** flag:

Exit Status

This command returns the following exit values:

0 Successful completion.

>0 An error occurred.

Examples

1. To list the static and external symbols of the object file `a.out`, enter:

   ```
   nm -e a.out
   ```

2. To display symbol sizes and values as hexadecimal and sort the symbols by value, enter:

   ```
   nm -xv a.out
   ```

3. To display symbol of all 64-bit objects in **libc.a**, ignoring all 32-bit objects:

   ```
   nm -X64 /usr/lib/libc.a
   ```

Files

/usr/ccs/bin/nm Contains the **nm** command.

Related Information

The **ar** command, **as** command, **ld** command, **size** command, **strip** command.

The **a.out** file, **ar** file.

Commands Overview in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices*.

---

no Command

Purpose

Manages network tuning parameters.

Syntax

no [**-p** | **-r**] { **-o** *Tunable*[**=***NewValue*] }

no [**-p** | **-r**] {**-d** *Tunable* }

no [**-p** | **-r**] { **-D** }

no -a

no -?

no -h *Tunable*

no -L *Tunable*

Note: Multiple flags **-o**, **-d**, and **-L** are allowed.

Description

Use the **no** command to configure network tuning parameters. The **no** command sets or displays current or next boot values for network tuning parameters. This command can also make permanent changes or defer changes until the next reboot. Whether the command sets or displays a parameter is determined by the accompanying flag. The **-o** flag performs both actions. It can either display the value of a parameter or set a new value for a parameter. When the **no** command is used to modify a network option it logs a message to the syslog using the LOG_KERN facility. For a more information on how the network parameters interact with each other, refer to the *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

Attention: Be careful when you use this command. If used incorrectly, the **no** command can cause your system to become inoperable.

Flags

-a Displays current, reboot (when used in conjunction with **-r**) or permanent (when used in conjunction with **-p**) value for all tunable parameters, one per line in pairs *Tunable* **=** *Value*. For the permanent options, a value only displays for a parameter if its reboot and current values are equal. Otherwise NONE displays as the value.

-d *Tunable* Resets *Tunable* its to default value. If *Tunable* needs to be changed (that is it is currently not set to its default value) and it is of type Bosboot or Reboot, or if it is of type Incremental and has been changed from its default value, and **-r** is not used in combination, it is not changed but a warning displays instead.

-D Resets all tunables to their default value. If a tunable needing to be changed are of type Bosboot or Reboot, or if they are of type Incremental and have been changed from their default value, and neither **-p** nor **-r** are used in combination, they will not be changed but a warning displayd instead.

-h *Tunable* Displays help about *Tunable* parameters.

-o *Tunable* [**=***NewValue*] Displays the value or sets the *Tunable* to *NewValue*. If a tunable needs to be changed (the specified value is different than current value), and is of type Bosboot or Reboot, or if it is of type Incremental and its current value is bigger than the specified value, and **-r** is not used in combination, it will not be changed but a warning displays instead.

When **-r** is used in combination without a new value, the nextboot value for *Tunable* is displayed. When **-p** is used in combination without a new value, a value displays only if the current and next boot values for tunable are the same Otherwise NONE displays as the value.

-p Makes changes apply to both current and reboot values when used in combination with **-o**, **-d** or **-D**, that is turns on updating of the **/etc/tunables/nextboot** file in addition to the updating of the current value. These combinations cannot be used on Reboot and Bosboot type parameters because their current value can't be changed.

When used with **-a** or *-o* without specifying a new value, values displays only if the current and next boot values for a parameter are the same. Otherwise NONE displays as the value.

-r Makes changes apply to reboot values when used in combination with **-o**, **-d** or **-D** , that is is turns on the updating of the **/etc/tunables/nextboot** file. If any parameter of type Bosboot is changed, the user is prompted to run bosboot. When used with **-a** or **-o** without specifying a new value, next boot values for tunables display instead of the current values.

-L *Tunable* Lsts the characteristics of one or all *Tunable*s, one per line, using the following format:

```
Name   Current  Default  Reboot   Minimum  Maximum  Unit  Type Dependencies
       value    value    value    value    value
-------------------------------------------------------------------------------
param1 5        2        4        1        10       MB/s     I  param2
                                                               param3
```

Parameters can be of the following types:

- **D** for Dynamic if tunable can be changed at any time
- **S** for Static if tunable can never be changed
- **R** for Reboot if tunable can only be changed during reboot sequence.
- **B** for Bosboot if tunable can only be changed by running bosboot and rebooting machine
- **M** for Mount if tunable is only effective for future filesystems or directory mountings
- **I** for Incremental if tunable can only be incremented, except at boot time

-? Displays the usage of the **no** command.

Any change (with **-o**, **-d**, or **-D**) to a parameter of type Mount results in a message sdisplaying to warn the user that the change is only effective for future mountings.

Any attempt to change (with **-o**, **-d**, or **-D**) a parameter of type Bosboot or Reboot without **-r**, results in an error message.

Any attempt to change (with **-o**, **-d**, or **-D** but without **-r**) the current value of a parameter of type Incremental with a new value smaller than the current value, results in an error message.

Note: The current set of parameters managed by the no command only includes Reboot, Static and Dynamic types.

Compatibility Mode

When running in pre 5.2 compatibility mode (controlled by the **pre520tune** attribute of **sys**0, see Introduction to AIX 5L Version 5.2 tunable parameter setting in the Performance Management Guide), reboot values for parameters, except those of type Bosboot, are not really meaningful because in this mode they are not applied at boot time.

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

This mode is automatically turned ON when a machine is MIGRATED to AIX 5L Version 5.2. For complete installations, it is turned OFF and the reboot values for parameters are set by applying the content of the **/etc/tunables/nextboot** file during the reboot sequence. Only in that mode are the **-r** and **-p** flags fully functional. See AIX 5L Version 5.2 kernel tuning in the Performance Tools Guide and Reference for details about the new 5.2 mode.

Network Tunable Parameters

arpqsize

Purpose:
Specifies the maximum number of packets to queue while waiting for ARP responses.

Value:
Default: 1
Range: 1 to MAXSHORT
Type: Dynamic

Diagnosis:
N/A

Tuning This parameter is supported by Ethernet, 802.3, Token-Ring and FDDI interfaces. The arpqsize value is increased to a minimum value of 5 when path MTU discovery is enabled. The value will not automatically decrease if path MTU discovery is subsequently disabled. This parameter applies to AIX 4.1.5, AIX 4.2.1 and later

Refer To:
Managing Tunable SP Parameters

arpt_killc

Purpose:
Specifies the time in minutes before a complete ARP entry will be deleted.

Value:
Default: 20
Range: 0 to MAXSHORT
Type: Dynamic

Diagnosis:
N/A

Tuning To reduce ARP activity in a stable network, you can increase **arpt_killc**.

Refer To:
SP System-Specific Tuning Recommendations

arptab_bsiz

Purpose:
Specifies Address Resolution Protocol (ARP) table bucket size.

Value:
Default: 7
Range: 1 to MAXSHORT
Type: Reboot

Diagnosis:
N/A

Tuning N/A

Refer To:
SP System-Specific Tuning Recommendations

arptab_nb

Purpose:
Specifies the number of ARP table buckets.

Value:
Default: 25
Range: 1 to MAXSHORT
Type: Reboot

Diagnosis:
N/A

Tuning Increase this value for systems that have a large number of clients or servers. The default provides for 25 x 7 = 175 ARP entries, but assumes an even task distribution.

Refer To:
SP System-Specific Tuning Recommendations

bcastping

Purpose:
Allows response to ICMP echo packets to the broadcast address.

Value:
Default: 0 (off)
Range: 0 or 1
Type: Dynamic

Diagnosis:
N/A

Tuning N/A

clean_partial_conns

Purpose:

Specifies whether or not SYN (synchronizes the sequence number) attacks are being avoided.

Value:

Default: 0 (off)

Range: 0 or 1

Type: Dynamic

Diagnosis:

N/A

Tuning This option should be turned on for servers that need to protect against network attacks. If on, randomly removes partial connections to make room for new non-attack connections.

delayack

Purpose:

Delays acknowlegements (ACK) for certain TCP packets and attempts to *piggyback* them with the next packet sent instead.

Value:

Default: 0

Range: 0 to 3

Type: Dynamic

Diagnosis:

N/A

Tuning This action will only be performed for connections whose destination port is specified in the list of the delayackports parameter. This can be used to increase performance when communicating with an HTTP server by reducing the total number of packets sent. The parameter can have one of four values:

- **0** No delays; normal operation
- **1** Delay the ACK for the server's SYN (Synchronizes the sequence numbers)
- **2** Delay the ACK for the server's FIN (Sender has reached the end of its byte stream)
- **3** Delay both the ACKs for the SYN and FIN.

delayackports

Purpose:

Specifies the list of destination ports for which the operation defined by the delayack port option will be performed.

Value:

Default: {}

Range: List of port numbers

Type: Dynamic

Diagnosis:

N/A

Tuning The parameter takes a list of up to ten ports, separated by commas and enclosed in curly braces. For example:

```
no -o delayackports={80,30080}
```

To clear the list, set the option to {}.

dgd_packets_lost

Purpose:
Specifies how many consecutive packets must be lost before Dead Gateway Detection decides that a gateway is down.

Value:
Default: 3
Range: 1 to MAXSHORT
Type: Dynamic

Diagnosis:
N/A

Tuning N/A

dgd_ping_time

Purpose:
Specifies how many seconds should pass between pings of a gateway by Active Dead Gateway Detection.

Value:
Default: 5
Range: 1 to MAXINT
Type: Dynamic

Diagnosis:
N/A

Tuning

dgd_retry_time

Purpose:
Specifies how many minutes a route's cost should remain raised when it has been raised by Passive Dead Gateway Detection. After this many minutes pass, the route's cost is restored to its user-configured value.

Value:
Default: 5
Range: 1 to MAXSHORT
Type: Dynamic

Diagnosis:
N/A

Tuning N/A

directed_broadcast

Purpose:
Specifies whether to allow a directed broadcast to a gateway.

Value:
Default: 1(on)
Range: 0 or 1
Type: Dynamic

Diagnosis:
N/A

Tuning The value of 1 allows packets to be directed to a gateway to be broadcast on a network on the other side of the gateway.

extendednetstats

Purpose:
Enables more extensive statistics for network memory services.

Value:

Default: 0(off)

Range: 0 or 1

Type: Reboot

Diagnosis:
N/A

Tuning The default for this parameter is 0, for off, because these extra statistics cause a reduction in system performance.

fasttimo

Purpose:
Allows you to set the millisecond delay for the TCP fast timeout timer. This timeout controls how often the system scans the TCP control blocks to send delayed acknowledgments.

Value:

Default: 200

Range: 50 to 200 milliseconds

Type: Dynamic

Diagnosis:
N/A

Tuning Reducing this timer value may improve performance with some non-IBM systems. However, this may also result in slightly increased system utilization.

icmp6_errmsg_rate

Purpose:
Specifies the upper limit for the number of ICMP v6 error messages that can be sent per second. This prevents excessive bandwidth being used by ICMP v6 error messages.

Value:

Default: 10 msg/sec

Range: 1 to 255

Type: Dynamic

Diagnosis:
N/A

Tuning N/A

icmpaddressmask

Purpose:
Specifies whether the system responds to an ICMP address mask request.

Value:

Default: 0(off)

Range: 0 or 1

Type: Dynamic

Diagnosis:
N/A

Tuning If the default value 0 is set, the network silently ignores any ICMP address mask request that it receives.

ie5_old_multicast_mapping

Purpose:
Specifies IP multicasts on Token-Ring should be mapped to the broadcast address rather than a functional address when value 1 is used.

Value:
Default: 0(off)
Range: 0 or 1
Type: Dynamic

Diagnosis:
N/A

Tuning N/A

ifsize

Purpose:
Specifies the maximum number of network interface structures per interface.

Value:
Default: Number of interfaces in the system
Range:
Type: Reboot

Diagnosis:
N/A

Tuning In AIX 4.3, the **/etc/rc.ne**t script automatically determines the number of interfaces on the system and sets ifsize to that value. Therefore, there is no reason to alter this parameter.

Refer To:
Managing Tunable SP Parameters

inet_stack_size

Purpose:
Specifies size of inet interrupt stack table in kilobytes.

Value:
Default: 16
Range: 1 to MAXSHORT
Type: Reboot

Diagnosis:
N/A

Tuning This is needed if you were running with unoptimized debug kernel or the netinet kernel extension. This is different from the pin more stack code (which is not configurable) because this is on interrupt. This parameter only needs to be changed if there is a system panic due to interrupt stack overflow.

ipforwarding

Purpose:
Specifies whether the kernel should forward packets.

Value:

Default: 0(off)

Range: 0 or 1

Type: Dynamic

Diagnosis:
N/A

Tuning Set this parameter to 1, if the system is acting as an IP router.

Refer To:
traceroute command

ipfragttl

Purpose:
Specifies the time to live for IP fragments in half-seconds.

Value:

Default: 60

Range: 60 to 255

Type: Dynamic

Diagnosis:
Fragments dropped after timeout (**netstat -p ip**).

Tuning If value of IP: fragments dropped after timeout is nonzero, increasing **ipfragttl** may reduce retransmissions.

Refer To:
IP Layer

ipignoreredirects

Purpose:
Specifies whether to process redirects that are received.

Value:

Default: 0(does redirects)

Range: 0 or 1(1 ignores redirects)

Type: Dynamic

Diagnosis:
N/A

Tuning This option only applies to AIX 4.2.1 or later.

ipqmaxlen

Purpose:
Specifies the number of received packets that can be queued on the IP protocol input queue.

Value:

Default: 100

Range: 100 to MAXINT

Type: Dynamic

Diagnosis:
Examine **ipintrq** overflows (**netstat -s**) or use crash to access IP input queue overflow counter.

Tuning Increase size if system is using a lot of loopback sessions. Most operating system network drives call IP directly and do not use the IP queue. On these devices increasing ipqmaxlen has no effect.

Refer To:
IP Layer and IP Protocol Performance Tuning Recommendations

ipsendredirects

Purpose:
Specifies whether the kernel should send redirect signals.

Value:

Default: 1(send redirects)

Range: 0(do not send redirects) or 1

Type: Dynamic

Diagnosis:
N/A

Tuning This is a configuration decision with performance consequences.

ipsrcrouteforward

Purpose:
Specifies whether the system forwards source-routed packets.

Value:

Default: 1(on)

Range: 0 or 1

Type: Dynamic

Diagnosis:
N/A

Tuning The default value of 1 allows the forwarding of source-routed packets. A value of 0 causes all source-routed packets that are not at their destinations to be discarded. This parameter only applies to AIX 4.2.1 or later.

ipsrcrouterecv

Purpose:
Specifies whether the system accepts source-routed packets.

Value:

Default: 0(off)

Range: 0 or 1

Type: Dynamic

Diagnosis:
N/A

Tuning The default value of 0 causes all source-routed packets destined for this system to be discarded. A value of 1 allows source-routed packets to be received. This parameter only applies to AIX 4.2.1 or later.

ipsrcroutesend

Purpose:

Specifies whether applications can send source-routed packets.

Value:

Default: 1(on)

Range: 0 or 1

Type: Dynamic

Diagnosis:

N/A

Tuning The default value of 1 allows source-routed packets to be sent. A value of 0 causes setsockopt() to return an error if an application attempts to set the source routing option, and removes any source routing options from outgoing packets. This parameter only applies to AIX 4.2.1 or later.

ip6_defttl

Purpose:

Specifies the default hop count that is used for Internet Protocol Version 6 (IPv6) packets if no other hop count is specified.

Value:

Default: 64

Range: 1 to 255

Type: Dynamic

Diagnosis:

N/A

Tuning N/A

ip6_prune

Purpose:

Specifies how often to check the IPv6 routing table for expired routes in seconds.

Value:

Default: 2

Range: 1 to MAXINT

Type: Dynamic

Diagnosis:

N/A

Tuning N/A

ip6forwarding

Purpose:

Specifies whether the kernel should forward IPv6 packets.

Value:

Default: 0(off)

Range: 0 or 1

Type: Dynamic

Diagnosis:

N/A

Tuning The default value of 0 prevents forwarding of IPv6 packets when they are not for the local systems. A value of 1 enables forwarding.

ip6srcrouteforward

Purpose:
Specifies whether the system forwards source-routed IPv6 packets.

Value:
Default: 1(on)
Range: 0 or 1
Type: Dynamic

Diagnosis:
N/A

Tuning The default value of 1 allows the forwarding of source-routed packets. A value of 0 causes all source-routed packets that are not at their destinations to be discarded.

llsleep_timeout

Purpose:
Specifies timeout value in seconds for link local timeouts (used when multi_homed=1)

Value:
Default: 3
Range: 1 to MAXINT
Type: Dynamic

Diagnosis:
N/A

Tuning N/A

main_if6

Purpose:
Specifies the interface to use for link local addresses. This is only used by **autoconf6** to set up initial routes.

Value:
Default: 0
Range: 0 or 1
Type: Dynamic

Diagnosis:
N/A

Tuning N/A

main_site6

Purpose:
Specifies the interface to use for site local address routing. This is only used if **multi_homed** is set to 3

Value:
Default: 0
Range: 0 or 1
Type: Dynamic

Diagnosis:
N/A

Tuning N/A

maxnip6q

Purpose:
Specifies the maximum number of IPv6 packet-reassembly queues.

Value:
Range: 1 to MAXSHORT
Type: Dynamic

Diagnosis:
N/A

Tuning N/A

maxttl

Purpose:
Specifies the time to live for Routing Information Protocol (RIP) packets in seconds.

Value:
Default: 255
Range: 1 to 255
Type: Dynamic

Diagnosis:
N/A

Tuning N/A

multi_homed

Purpose:
Specifies the level of multi-homed IPv6 host support.

Value:
Default: 1
Range: 0 to 3
Type: Dynamic

Diagnosis:
N/A

Tuning This will only be performed for connections whose destination port is specified in the list of the delayackports parameter. This can be used to increase performance when communicating with an HTTP server. The parameter can have one of four values:

- **0** - Indicates the original functionality in AIX 4.3.
- **1** - Indicates that link local addresses will be resolved by querying each interface for the link local address.
- **2** - Indicates that link local addresses will only be examined for the interface defined by **main_if6**.
- **3** - Indicates that link local addresses will only be examined for the interface defined by **main_if6**, and site local addresses will only be routed for the main_site6 interface.

nbc_limit

Purpose:

Specifies the total maximum amount of memory that can be used for the Network Buffer Cache (NBC) in Kbytes.

Value:

Default: derived from thewall

Range: 0 to MAXINT

Type: Dynamic

Diagnosis:

N/A

Tuning When the cache grows to this limit, the least-used caches are flushed out of cache to make room for the new ones. This parameter only applies to AIX 4.3.2 or later. NBC is only used by the **send_file**() API and some Web servers that use the get engine in the kernel.

nbc_max_cache

Purpose:

Specifies the maximum size of the cache object allowed in the Network Buffer Cache (NBC) in bytes.

Value:

Default: 131072(128K)

Range: 1 to MAXINT

Type: Dynamic

Diagnosis:

N/A

Tuning Data object bigger than this size is not be put in the NBC. This parameter only applies to AIX 4.3.2 or later. NBC is only used by the **send_file**() API and some Web servers that use the get engine in the kernel.

nbc_min_cache

Purpose:

Specifies the minimum size of the cache object allowed in the Network Buffer Cache (NBC) in bytes.

Value:

Default: 1 byte

Range: 1 to 131072(128K)

Type: Dynamic

Diagnosis:

N/A

Tuning Data object smaller than this size is not be put in the NBC. This parameter only applies to Version 4.3.2 or later. NBC is only used by the **send_file**() API and some Web servers that use the get engine in the kernel

nbc_pseg (AIX 4.3.3 and later)

Purpose:

Specifies the maximum number of private segments that can be created for the Network Buffer Cache (NBC).

Value:

Default: 0

Range: 0 to MAXSHORT

Type: Dynamic

Diagnosis:

N/A

Tuning When this option is set to a nonzero value, data objects with size between the size specified in **nbc_max_cache** and the segment size (256 MB) will be cached in a private segment. Data objects bigger than the segment size will not be cached at all. When this many private segments exist in NBC, cache data in private segments may be flushed for new cache data so the number of private segments will not exceed the limit. When this option is set to 0, all cache in private segments will be flushed.

nbc_pseg_limit (AIX 4.3.3 and later)

Purpose:

Specifies the maximum total cached data size (KB) allowed in private segments in the Network Buffer Cache (NBC).

Value:

Default: 131072

Range: 0 to MAXSHORT

Type: Dynamic

Diagnosis:

N/A

Tuning Because data cached in private segments will be pinned by the Network Buffer Cache, this option provides a control on the amount of pinned memory used for Network Buffer Cache in addition to the network buffers in global segments. When this limit is met, cache data in private segments may be flushed for new cache data so the total pinned memory size will not exceed the limit. When this option is set to 0, all cache in private segments will be flushed.

ndpqsize

Purpose:

Specifies the number of packets to hold waiting on completion of a Neighbor Discovery Protocol (NDP) entry (used by MTU Path Discovery).

Value:

Default: 50

Range: 1 to MAXSHORT

Type: Dynamic

Diagnosis:

N/A

Tuning N/A

ndpt_down

Purpose:

Specifies the time, in half-seconds, to hold down a Neighbor Discovery Protocol (NDP) entry.

Value:

Default: 3 (1.5 seconds)

Range: 1 to MAXINT

Type: Dynamic

Diagnosis:

N/A

Tuning N/A

ndpt_keep

Purpose:

Specifies the time, in half seconds, to keep a Neighbor Discovery Protocol (NDP) entry.

Value:

Default: 120 (60 seconds)

Range: 1 to MAXINT

Type: Dynamic

Diagnosis:

N/A

Tuning N/A

ndp_mmaxtries

Purpose:

pecifies the maximum number of Multicast NDP packets to send.

Value:

Default: 3

Range: 0 to MAXINT

Type: Dynamic

Diagnosis:

N/A

Tuning N/A

ndpt_probe

Purpose:

Specifies the time, in half-seconds, to delay before sending the first Neighbor Discovery Protocol (NDP) probe.

Value:

Default: 5 (2.5 seconds)

Range: 1 to MAXINT

Type: Dynamic

Diagnosis:

N/A

Tuning N/A

ndpt_reachable

Purpose:

Specifies the time, in half-seconds, to test if a Neighbor Discovery Protocol (NDP) entry is still valid.

Value:

Default: 30 (15 seconds)

Range: 1 to MAXINT

Type: Dynamic

Diagnosis:

N/A

Tuning N/A

ndpt_retrans

Purpose:

Specifies the time, in half-seconds, to wait before retransmitting a Neighbor Discovery Protocol (NDP) request.

Value:

Default: 1(half a second)

Range: 1 to MAXINT

Type: Dynamic

Diagnosis:

N/A

Tuning N/A

ndpt_umaxtries

Purpose:

Specifies the maximum number of Unicast NDP packets to send.

Value:

Default: 3

Range: 0 to MAXINT

Type: Dynamic

Diagnosis:

N/A

Tuning N/A

net_malloc_police

Purpose:

Specifies the size of the **net_malloc** and **net_free** trace buffers.

Value:

Default: 0

Range: 0 to MAXINT

Type: Dynamic

Diagnosis:

N/A

Tuning If the value of this variable is non-zero, all **net_malloc** and **net_free** buffers will be traced in a kernel buffer and by system trace hook HKWD_NET_MALLOC. Additional error-checking will also be enabled. This includes checks for freeing a free buffer, alignment, and buffer overwrite. Enable this parameter only when investigating some network problem, because performance is affected negatively when turned on. The default value is zero (policing off). Values of **net_malloc_police** larger than 1024 will allocate that many items in the kernel buffer for tracing.

nonlocsrcroute

Purpose:

Tells the Internet Protocol that strictly source-routed packets may be addressed to hosts outside the local network.

Value:

Default: 0

Range: 0 or 1

Type: Dynamic

Diagnosis:

N/A

Tuning A default value of 0 disallows addressing to outside hosts. The value of 1 allows packets to be addressed to outside hosts. Loosely source-routed packets are not affected by this parameter. This is a configuration decision with minimal performance consequences.

passive_dgd

Purpose:

Specifies whether Passive Dead Gateway Detection is enabled. A value of 0 turns it off, and a value of 1 enables it for all gateways in use.

Value:

Default: 0

Range: 0 or 1

Type: Dynamic

Diagnosis:

N/A

Tuning N/A

pmtu_default_age

Purpose:

Specifies the default amount of time (in minutes) before the path MTU value for UDP paths is checked for a lower value.

Value:

Default: 10

Range: 0 to MAXINT

Type: Dynamic

Diagnosis:

N/A

Tuning A value of zero allows no aging. The default value is 10 minutes. The **pmtu_default_age** value can be overridden by UDP applications. This parameter only applies to AIX 4.2.1 or later.

pmtu_rediscover_interval

Purpose:

Specifies the default amount of time (in minutes) before the path MTU value for UDP and TCP paths are checked for a higher value.

Value:

Default: 30

Range: 0 to MAXINT

Type: Dynamic

Diagnosis:

N/A

Tuning A value of 0 allows no path MTU rediscovery. The default value is 30 minutes. This parameter only applies to AIX 4.2.1 or later.

rfc1122addrchk

Purpose:

Performs address validation as specified by RFC1122 (Requirements for Internet Hosts-Communication Layers).

Value:

Default: 0(off)

Range: 0 or 1

Type: Dynamic

Diagnosis:

N/A

Tuning The default value of 0 does not perform address validation. A value of 1 performs address validation.

rfc1323

Purpose:

Enables TCP enhancements as specified by RFC 1323 (TCP Extensions for High Performance). Value of 1 indicates that **tcp_sendspace** and **tcp_recvspace** can exceed 64 KB.

Value:

Default: 0(off)

Range: 0 or 1

Type: Dynamic

Diagnosis:

N/A

Tuning The default value of 0 disables the RFC enhancements on a systemwide scale. A value of 1 specifies that all TCP connections will attempt to negotiate the RFC enhancements. The SOCKETS application can override the default behavior on individual TCP connections, using the **setsockopt** subroutine. Make changes before attempting to set **tcp_sendspace** and **tcp_recvspace** to more than 64 KB.

Refer to:

TCP Socket Buffer Tuning

rfc2414

Purpose:

Enables the increasing of TCP's initial window as described in RFC 2414. When it is on, the initial window will depend on the setting of the tunable **tcp_init_window**.

Value:

Default: 0 (off)

Range: 0 or 1 (on)

Type: Dynamic

Diagnosis:

N/A

Tuning N/A

route_expire

Purpose:

Specifies whether the route expires.

Value:

Default: 0(off)

Range: 0 or 1

Type: Dynamic

Diagnosis:

N/A

Tuning A value of 0 allows no route expiration, which is the default. Negative values are not allowed for this option. This parameter only applies to Version 4.2.1 or later.

Refer to:

Tuning TCP Maximum Segment Size

routerevalidate

Purpose:

Specifies that each connection's cached route should be revalidated each time a new route is added to the routing table. This will ensure that applications that keep the same connection open for long periods of time (for example, NFS) use the correct route after routing-table changes occur.

Value:

Default: 0(off)

Range: 0 or 1

Type: Dynamic

Diagnosis:

If there is communication loss between HACMP nodes or between NFS client and server, routes may have been changed but the values in the cache are being used.

Tuning The default value of 0 does not revalidate the cached routes. Turning on this option may cause some performance degradation. Value should be set to 1 for HACMP.

rto_high

Purpose:

Specifies the TCP Retransmit timeout high value used in calculating factors and the maximum retransmittals allowable used in TCP data segment retransmittals.

Value:

Default: 64

Range: 2 to MAXINT

Type: Dynamic

Diagnosis:

N/A

Tuning The **rto_high** parameter is the high factor.

rto_length

Purpose:

Specifies the TCP Retransmit timeout length value used in calculating factors and the maximum retransmittals allowable used in TCP data segment retransmittals.

Value:

Default: 13

Range: 2 to MAXINT

Type: Dynamic

Diagnosis:

N/A

Tuning The **rto_length** parameter is the total number of time segments.

rto_limit

Purpose:

Specifies the TCP Retransmit timeout limit value used in calculating factors and the maximum retransmittals allowable used in TCP data segment retransmittals.

Value:

Default: 7

Range: 1 to 64

Type: Dynamic

Diagnosis:

N/A

Tuning The **rto_limit** parameter is the number of time segments from **rto_low** to **rto_high**.

rto_low

Purpose:

Specifies the TCP Retransmit timeout low value used in calculating factors and the maximum retransmittals allowable used in TCP data segment retransmittals.

Value:

Default: 1

Range: 0 to 63

Type: Dynamic

Diagnosis:

N/A

Tuning The **rto_low** parameter is the low factor.

sack (AIX 4.3.3 and later)

Purpose:

Specifies if Selective Acknowledgment is on.

Value:

Default: 0(off)

Range: 0 or 1

Type: Dynamic

Diagnosis:

N/A

Tuning Normal TCP has to retransmit all packets after a dropped packet. With large window sizes, this can result in lost performance if many packets are dropped. Selective Ack enables code to selectively retransmit the lost packet(s). If there are a lot of retransmits of fragments and the receive buffer sizes are large, then it might be beneficial to turn this parameter on. This is a TCP negotiated option, so it must be supported on both endpoints before it will be used.

sb_max

Purpose:

Specifies the maximum buffer size allowed for a TCP and UDP socket. Limits **setsockopt**, **udp_sendspace**, **udp_recvspace**, **tcp_sendspace**, and **tcp_recvspace**.

Value:

Default: 1048576 bytes

Range: 1 to MAXINT

Type: Dynamic

Diagnosis:

N/A

Tuning Increase size, preferably to multiple of 4096. Should be approximately two to four times the largest socket buffer limit.

Refer to:

TCP Socket Buffer Tuning and Tuning Other Layers to Improve NFS Performance

send_file_duration

Purpose:

Specifies the cache validation duration for all the file objects that the **send_file** system call accessed in the Network Buffer Cache (in seconds).

Value:

Default: 300 (5 minutes)

Range: 0 to MAXINT

Type: Dynamic

Diagnosis:

N/A

Tuning A value of 0 means that the cache will be validated for every access. This parameter only applies to AIX 4.3.2 or later.

site6_index

Purpose:
Specifies the maximum interface number for site local routing.

Value:
Default: 0
Range: 0 to MAXSHORT
Type: Dynamic

Diagnosis:
N/A

Tuning N/A

sockthresh

Purpose:
Specifies the maximum amount of network memory that can be allocated for sockets. Used to prevent new sockets or TCP connections from exhausting all MBUF memory and reserve the remaining memory for the existing sockets or TCP connections.

Value:
Default: 85 percent
Range: 0 to 100
Type: Dynamic

Diagnosis:
N/A

Tuning When the total amount of memory allocated by the **net_malloc** subroutine reaches this threshold, the **socket** and **socketpair** system calls fail with an error of ENOBUFS. Incoming connection requests are silently discarded. Existing sockets can continue to use additional memory. The sockthresh parameter represents a percentage of the thewall parameter, with possible values of 1 to 100 and a default of 85.

Refer to:
Socket Layer

sodebug

Purpose:
Specifies whether the newly created sockets will have SO_DEBUG flag on.

Value:
Default: 0 (no)
Range: 0 or 1 (yes)
Type: Dynamic

Diagnosis:
N/A

Tuning N/A

somaxconn

Purpose:
Specifies the maximum listen backlog.

Value:
Default: 1024 connections
Range: 0 to MAXSHORT
Type: Dynamic

Diagnosis:
N/A

Tuning Increase this parameter on busy Web servers to handle peak connection rates.

subnetsarelocal

Purpose:

Specifies whether all subnets that match the subnet mask are to be considered local for purposes of establishing, for example, the TCP maximum segment size.

Value:

Default: 1(yes)

Range: 0 or 1

Type: Dynamic

Diagnosis:

N/A

Tuning This parameter is used by the **in_localaddress** subroutine. The default value of 1 specifies that addresses that match the local network mask are local. If the value is 0, only addresses matching the local subnetwork are local. This is a configuration decision with performance consequences. If the subnets do not all have the same MTU, fragmentation at bridges may degrade performance. If the subnets do have the same MTU, and **subnetsarelocal** is 0, TCP sessions may use an unnecessarily small MSS.

Refer to:

Tuning TCP Maximum Segment Size

tcp_bad_port_limit

Purpose:

Specifies the number of TCP packets to a port with no socket that can be received in a 500 millisecond period before TCP stops sending resets in response to such packets. When it is set as its default value 0, resets will always be sent when TCP packets are received for a bad port number.

Value:

Default: 0

Range: 1 to MAXINT

Type: Dynamic

Diagnosis:

N/A

Tuning N/A

tcp_ecn

Purpose:

Enables TCP level support for Explicit Congestion Notification as described in RFC 2481. Default is off (0). Turning it on (1) will make all connections negotiate ECN capability with the peer. For this feature to work you need support from the peer TCP and also IP level ECN support from the routers in the path.

Value:

Default: 0 (off)

Range: 0 or 1 (on)

Type: Dynamic

Diagnosis:

N/A

Tuning N/A

tcp_ephemeral_high

Purpose:
Specifies the largest port number to allocate for TCP ephemeral ports.

Value:

Default: 65535

Range: 32769 to 65535

Type: Dynamic

Diagnosis:
N/A

Tuning The number of ephemeral sockets is determined by **tcp_ephemeral_high** minus tcp_ephemeral_low. For maximum number of ephemeral sockets, set **tcp_ephemeral_high** to 65535 and **tcp_ephemeral_low** to 1024.

Refer to:
Socket Layer

tcp_ephemeral_low

Purpose:
Specifies the smallest port number to allocate for TCP ephemeral ports.

Value:

Default: 32768

Range: 1024 to 65534

Type: Dynamic

Diagnosis:
N/A

Tuning The number of ephemeral sockets is determined by **tcp_ephemeral_high** minus **tcp_ephemeral_low**. For maximum number of ephemeral sockets, set **tcp_ephemeral_high** to 65535 and **tcp_ephemeral_low** to 1024.

Refer to:
Socket Layer

tcp_init_window

Purpose:
This value is used only when rfc2414 is turned on (ignored otherwise). If rfc2414 is on and this value is zero, then the initial window computation is done according to rfc2414. If this value is non-zero, the initial(congestion) window is initialized a number of maximum sized segments equal to **tcp_init_window**.

Value:

Default: 0

Range: 0 to MAXSHORT

Type: Dynamic

Diagnosis:
N/A

Tuning N/A

tcp_keepcnt

Purpose:

Represents the number of keepalive probes that could be sent before terminating the connection.

Value:

Default: 8

Range: 0 to MAXINT

Type: Dynamic

Diagnosis:

N/A

Tuning

tcp_keepidle

Purpose:

Specifies the length of time to keep an idle TCP connection active, measured in half-seconds.

Value:

Default: 14400 (2 hours)

Range: 1 to MAXINT

Type: Dynamic

Diagnosis:

N/A

Tuning This is a configuration decision with minimal performance consequences. No change is recommended.

tcp_keepinit

Purpose:

Sets the initial timeout value for a TCP connection in half-seconds.

Value:

Default: 150 (75 seconds)

Range: 1 to MAXINT

Type: Dynamic

Diagnosis:

N/A

Tuning N/A

tcp_keepintvl

Purpose:

Specifies the interval, measured in half-seconds, between packets sent to validate the TCP connection.

Value:

Default: 150 (75 seconds)

Range: 1 to MAXSHORT

Type: Dynamic

Diagnosis:

N/A

Tuning This is a configuration decision with minimal performance consequences. No change is recommended. If the interval were shortened significantly, processing and bandwidth costs might become significant.

tcp_limited_transmit

Purpose:

Enables the feature that enhances TCP's loss recovery as described in the RFC 3042.

Value:

Default: 1 (on)

Range: 0 or 1

Type: Dynamic

Diagnosis:

N/A

Tuning N/A

tcp_maxburst

Purpose:

Specifies the number of back-to-back packets that TCP can send before pausing to allow those packets to be forwarded to their destination. This can be useful if routers are unable to handle large bursts of TCP packets and are dropping some of them. A value of 0 means no limitation for back-to-back packets before pausing.

Value:

Default: 0

Range: 0 to MAXSHORT

Type: Dynamic

Diagnosis:

N/A

Tuning N/A

tcp_mssdflt

Purpose:

Default maximum segment size used in communicating with remote networks.

Value:

Default: 512 bytes

Range: 1 to 1448

Type: Dynamic

Diagnosis:

N/A

Tuning For AIX 4.2.1 or later, **tcp_mssdflt** is only used if path MTU discovery is not enabled or path MTU discovery fails to discover a path MTU. Limiting data to (MTU - 52) bytes ensures that, where possible, only full packets will be sent.

Refer to:

Tuning TCP Maximum Segment Size

tcp_nagle_limit

Purpose:

This is the Nagle Algorithm threshold in bytes which can be used to disable Nagle. The default (65535 - the maximum size of IP packet) is Nagle turned on. To disable Nagle, set this value to 0 or 1. TCP disables Nagle for data segments larger than or equal to this threshold value.

Value:

Default: 65535 - maximum size of IP packet

Range: 0 to 65535

Type: Dynamic

Diagnosis:

N/A

Tuning N/A

tcp_ndebug

Purpose:

Specifies the number of tcp_debug structures.

Value:

Default: 100

Range: 0 to MAXSHORT

Type: Dynamic

Diagnosis:

N/A

Tuning N/A

tcp_newreno

Purpose:

Enables the modification to TCP's Fast Recovery algorithm as described in RFC 2582. This fixes the limitation of TCP's Fast Retransmit algorithm to recover fast from dropped packets when multiple packets in a window are dropped. sack also achieves the same thing but sack needs support from both ends of the TCP connection; the NewReno modification is only on the sender side.

Value:

Default: 1 (on)

Range: 0 or 1

Type: Dynamic

Diagnosis:

N/A

Tuning N/A

tcp_nodelayack

Purpose:
Specifies that sockets using TCP follow the Nagle algorithm when sending data.

Value:
Default: 0(no)
Range: 0 or 1
Type: Dynamic

Diagnosis:
N/A

Tuning This is an Interface-Specific Network Option (ISNO) option.

Refer to:
Relative Level of Function in UDP and TCP and Interface-Specific Network Options (ISNO)

tcp_pmtu_discover

Purpose:
Enables or disables path MTU discovery for TCP applications.

Value:
Default: 0 (1 in AIX 4.3.3)
Range: 0 or 1
Type: Dynamic

Diagnosis:
N/A

Tuning A value of 0 disables path MTU discovery for TCP applications, while a value of 1 enables it. This parameter only applies to AIX 4.2.1 or later.

Refer to:
Tuning TCP Maximum Segment Size

tcp_recvspace

Purpose:
Specifies the system default socket buffer size for receiving data. This affects the window size used by TCP.

Value:
Default: 16384 bytes
Range: 4096 to 1048576
Type: Dynamic

Diagnosis:
N/A

Tuning Setting the socket buffer size to 16 KB (16,384) improves performance over standard Ethernet and Token-Ring networks. Lower bandwidth networks, such as Serial Line Internet Protocol (SLIP), or higher bandwidth networks, such as Serial Optical Link, should have different optimum buffer sizes. The optimum buffer size is the product of the media bandwidth and the average round-trip time of a packet. The **tcp_recvspace** parameter must specify a socket buffer size less than or equal to the setting of the sb_max parameter. This is a Dynamic parameter, but for daemons started by the **inetd** daemon, run the following commands: `stopsrc -s inetd ; startsrc -s inetd`

Refer to:
TCP Socket Buffer Tuning

tcp_sendspace

Purpose:

Specifies the system default socket buffer size for sending data.

Value:

Default: 16384 bytes

Range: 4096 to 1048576

Type: Dynamic

Diagnosis:

N/A

Tuning This affects the window size used by TCP. Setting the socket buffer size to 16 KB (16,384) improves performance over standard Ethernet and Token-Ring networks.

Lower bandwidth networks, such as Serial Line Internet Protocol (SLIP), or higher bandwidth networks, such as Serial Optical Link, should have different optimum buffer sizes. The optimum buffer size is the product of the media bandwidth and the average round-trip time of a packet:

```
optimum_window=bandwidth * average_round_trip_time
```

The **tcp_sendspace** parameter must specify a socket buffer size less than or equal to the setting of the sb_max parameter. The **tcp_sendspace** parameter is a Dynamic parameter, but for daemons started by the inetd daemon, run the following commands:

```
stopsrc -s inetd ; startsrc -s inetd
```

Refer to:

TCP Socket Buffer Tuning

tcp_timewait

Purpose:

The **tcp_timewait** option is used to configure how long connections are kept in the timewait state in 15-second intervals.

Value:

Default: 1

Range: 1 to 5

Type: Dynamic

Diagnosis:

N/A

Tuning Increasing this value will degrade performance of Web servers or applications that open and close a lot of TCP connections.

tcp_ttl

Purpose:

Specifies the time to live for TCP packets.

Value:

Default: 60 ticks (100 ticks per minute)

Range: 1 to 255

Type: Dynamic

Diagnosis:

netstat -s

Tuning If the system is experiencing TCP timeouts, increasing **tcp_ttl** may reduce retransmissions.

thewall

Purpose:

Specifies the maximum amount of memory, in kilobytes, that is allocated to the memory pool.

Value:

Default: AIX 5.1 and later: 1/2 of RAM or 6.5 GB (whichever is smaller)

Range: N/A

Type: Static

Diagnosis:

N/A

Tuning Not settable from AIX 5.1 and later.

Refer to:

Tuning mbuf Pool Performance

udp_bad_port_limit

Purpose:

Specifies the number of UDP packets to a port with no socket that can be received in a 500 millisecond period before UDP stops sending ICMP errors in response to such packets. If set to 0, ICMP errors will always be sent when UDP packets are received for a bad port number. If greater than 0, it specifies the number of packets to be received before UDP stops sending ICMP errors.

Value:

Default: 0

Range: 0 to MAXINT

Type: Dynamic

Diagnosis:

N/A

Tuning N/A

udp_ephemeral_high

Purpose:

Specifies the largest port number to allocate for UDP ephemeral ports.

Value:

Default: 65535

Range: 32769 to 65535

Type: Dynamic

Diagnosis:

N/A

Tuning N/A

Refer to:

Socket Layer

udp_ephemeral_low

Purpose:
Specifies the smallest port number to allocate for UDP ephemeral ports.

Value:

Default: 32768

Range: 1 to 65534

Type: Dynamic

Diagnosis:
N/A

Tuning N/A

Refer to:
Socket Layer

udp_pmtu_discover

Purpose:
Enables or disables path MTU discovery for UDP applications.

Value:

Default: 0 (1 in AIX 4.3.3)

Range: 0 or 1

Type: Dynamic

Diagnosis:
N/A

Tuning UDP applications must be specifically written to use path MTU discovery. A value of 0 disables the feature, while a value of 1 enables it. The default value is 0. This parameter only applies to AIX 4.2.1 or later.

udp_recvspace

Purpose:
Specifies the system default socket-buffer size for receiving UDP data.

Value:

Default: 42080 bytes

Range: 4096 to 1048576

Type: Dynamic

Diagnosis:
Nonzero *n* in **netstat -s** report of udp: *n* socket buffer overflows

Tuning The **udp_recvspace** parameter must specify a socket buffer size less than or equal to the setting of the **sb_max** parameter. Increase size, preferably to multiple of 4096.

Refer to:
UDP Socket Buffer Tuning

udp_sendspace

Purpose:
Specifies the system default socket-buffer size for sending UDP data.

Value:
Default: 9216 bytes
Range: 4096 to 1048576
Type: Dynamic

Diagnosis:
N/A

Tuning The **udp_sendspace** parameter must specify a socket buffer size less than or equal to the setting of the **sb_max** parameter. Increase size, preferably to multiple of 4096.

Refer to:
UDP Socket Buffer Tuning

udp_ttl

Purpose:
Specifies the time to live for UDP packets.

Value:
- Default: 30 second
- Range: 1 to 255
- Type: Dynamic

Diagnosis:
N/A

Tuning N/A

udpcksum

Purpose:
Allows UDP checksum to be turned on/off.

Value:
Default: 1
Range: 0 or 1
Type: Dynamic

Diagnosis:
N/A

Tuning A value of 0 turns it off; while a value of 1 turns it on.

use_isno (AIX 4.3.3 and later)

Purpose:
Allows per interface tuning options.

Value:
Default: 1(on)
Range: 0 or 1
Type: Dynamic

Diagnosis:
N/A

Tuning A value of 1 means it is on, 0 is off. If the TCP tunable parameters per interface (tunable through SMIT or the chdev command) have been set, they will override the TCP global values if use_isno is set to 1. Application can still override all of these with the **setsockopt** subroutine.

Refer to:
Interface-Specific Network Options (ISNO)

Streams Tunable Parameters (AIX 4.2 or later)

lowthresh

Purpose:

Specifies the maximum number of bytes that can be allocated using the **allocb** call for the BPRI_LO priority.

Value:

Default: 90 (percent of thewall)

Range: 0 to 100

Type: Dynamic

Diagnosis:

N/A

Tuning When the total amount of memory allocated by the **net_malloc** call reaches this threshold, then the **allocb** request for the BPRI_LO priority returns 0. The **lowthresh** parameter represents a percentage of the thewall parameter, and you can set its value from 0 to 100.

medthresh

Purpose:

Specifies the maximum number of bytes that can be allocated using the allocb() call for the BPRI_MED priority.

Value:

Default: 95 (percent of thewall)

Range: 0 to 100

Type: Dynamic

Diagnosis:

N/A

Tuning When the total amount of memory allocated by the **net_malloc** call reaches this threshold, then the **allocb** request for the BPRI_MED priority returns 0. The **medthresh** parameter represents a percentage of the thewall parameter, and you can set its value from 0 to 100.

nstrpush

Purpose:

Specifies the maximum number (should be at least 8) of modules that you can push onto a single stream.

Value:

Default: 8

Range: 8 to MAXSHORT

Type: Dynamic

Diagnosis:

N/A

Tuning N/A

psebufcalls

Purpose:
Specifies the maximum number of bufcalls to allocate by streams.

Value:

Default: 20

Range: 20 to MAXINT

Type: Dynamic

Diagnosis:
N/A

Tuning The stream subsystem allocates certain number of bufcall structures at initialization. When the **allocb** call fails, the user can register their requests for the **bufcall** call. You cannot lower this value until the system reboots, at which time it returns to its default value.

psecache

Purpose:
Controls the number of stream buffers.

Value:

Default: 1

Range: 0 or 1

Type: Dynamic

Diagnosis:
N/A

Tuning N/A

pseintrstack

Purpose:
Specifies the maximum size of the interrupt stack allowed by streams while running in the offlevel.

Value:

Default: 0x3000 (decimal 12288)

Range: 12288 to MAXINT

Type: Reboot

Diagnosis:
N/A

Tuning When a process running other than INTBASE level enters into a stream, it occasionally encounters a stack overflow problem because the interrupt stack size is too small. Setting this parameter correctly reduces the chances of stack overflow problems.

psetimers

Purpose:
Specifies the maximum number of timers to allocate by streams.

Value:

Default: 20

Range: 20 to MAXINT

Type: Dynamic

Diagnosis:
N/A

Tuning The stream subsystem allocates a certain number of timer structures at initialization, so that the streams driver or module can register their **timeout** calls. You cannot lower this value until the system reboots, at which time it returns to its default value.

strctlsz

Purpose:
Specifies the maximum number of bytes of information that a single system call can pass to a stream to place into the control part of a message (in an M_PROTO or M_PCPROTO block).

Value:

Default: 1024

Range: 1 to MAXSHORT

Type: Dynamic

Diagnosis:
N/A

Tuning A **putmsg** call with a control part exceeding this size will fail with ERANGE.

strmsgsz

Purpose:
Specifies the maximum number of bytes of information that a single system call can pass to a stream to place into the data part of a message (in M_DATA blocks).

Value:

Default: 0

Range: 1 to MAXSHORT

Type: Dynamic

Diagnosis:
N/A

Tuning Any **write** call exceeding this size is broken into multiple messages. A **putmsg** call with a data part exceeding this size will fail with ERANGE.

strthresh

Purpose:

Specifies the maximum number of bytes that streams are normally allowed to allocate, expressed as a percentage of thewall.

Value:

Default: depends on **thewall**

Range: 0 to 100

Type: Dynamic

Diagnosis:

N/A

Tuning When the threshold is exceeded, it does not allow users without the appropriate privilege to open streams, push modules, or write to streams devices, and returns ENOSR. The threshold applies only to output side and does not affect data coming into the system (for example, the console continues to work correctly). A value of zero means that there is no threshold. The **strthresh** parameter represents a percentage of the **thewall** parameter, and you can set its value from 0 to 100. The **thewall** parameter indicates the maximum number of bytes that can be allocated by streams and sockets using the **net_malloc** call. When you change **thewall** parameter, the threshold is updated accordingly.

Refer to:

Tuning Network Memory

strturncnt

Purpose:

Specifies the maximum number of requests handled by the current running thread for Module- or Elsewhere-level streams synchronization.

Value:

Default: 15

Range: 1 to MAXINT

Type: Dynamic

Diagnosis:

N/A

Tuning With Module-level synchronization, only one thread can run in the module at any time, and all other threads which try to acquire the same module will enqueue their requests and exit. After the current running thread completes its work, it dequeues all the previously enqueued requests one by one and invokes them. If there is a large number of requests enqueued in the list, then the current running thread has to serve everyone and will always be busy serving others and starves itself. To avoid this situation the current running thread serves only the strturncnt number of threads, following that, a separate kernel thread wakes up and invokes all the pending requests.

Examples

1. To display the maximum size of the mbuf pool, type:

   ```
   no -o thewall
   ```

2. To reset the time to live for UDP packets its default size, type:

   ```
   no -d udp_ttl
   ```

3. To change the default socket buffer sizes on your system, type:

   ```
   no -r -o tcp_sendspace=32768
   no -r -o udp_recvspace=32768
   ```

4. To use a machine as an internet work router over TCP/IP networks, type:

   ```
    no -o ipforwarding=1
   ```

5. To list the current and reboot value, range, unit, type and dependencies of all tunables parameters managed by the **no** command, type:

   ```
   no -L
   ```

6. To display help information on udp_ephemeral_high, type:

   ```
   no -h udp_ephemeral_high
   ```

7. To permanently turn off ip6srcrouteforward, type:

   ```
   no -p -o ip6srcrouteforward=0
   ```

8. To list the reboot values for all Network tuning parameters, type:

   ```
   no -r -a
   ```

Related Information

The **ifconfig** command, the **inetd** command, the **vmo** command, the **ioo** command, the **schedo** command, the **tunsave** command, the **tunrestore** command, the **tuncheck** command, the **tundefault** command.

The **setsockopt** subroutine.

Network Overview for System Management in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

TCP/IP Addressing, TCP/IP Protocols, and TCP/IP Routing, Internet Protocol (IP), Transmission Control Protocol (TCP), Internet Application-Level Protocols, and User Datagram Protocol (UDP) in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

Monitoring and Tuning Communications I/O Use in *AIX 5L Version 5.2 Performance Management Guide*.

Path MTU Discovery in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

Kernel Tuning in *AIX 5L Version 5.2 Performance Tools Guide and Reference*.

---

nohup Command

Purpose

Runs a command without hangups.

Syntax

nohup *Command* [*Arg* ...] [**&**]

Description

The **nohup** command runs the command specified by the *Command* parameter and any related *Arg* parameters, ignoring all hangup (**SIGHUP**) signals. Use the **nohup** command to run programs in the background after logging off. To run a **nohup** command in the background, add an & (ampersand) to the end of the command.

Whether or not the **nohup** command output is redirected to a terminal, the output is appended to the **nohup.out** file in the current directory. If the **nohup.out** file is not writable in the current directory, the output is redirected to the **$HOME/nohup.out** file. If neither file can be created nor opened for appending, the command specified by the *Command* parameter is not invoked. If the standard error is a terminal, all output written by the named command to its standard error is redirected to the same file descriptor as the standard output.

Exit Status

This command returns the following exit values:

126 The command specified by the *Command* parameter was found but could not be invoked.

127 An error occurred in the **nohup** command or the command specified by the *Command* parameter could not be found.

Otherwise, the exit status of the **nohup** command is that of the command specified by the *Command* parameter.

Examples

1. To run a command in the background after you log off, enter:

   ```
   $ nohup find / -print &
   ```

   After you enter this command, the following is displayed:

   ```
   670
   $ Sending output to nohup.out
   ```

   The process ID number changes to that of the background process started by & (ampersand). The message `Sending output to nohup.out` informs you that the output from the **find / -print** command is in the **nohup.out** file. You can log off after you see these messages, even if the **find** command is still running.

2. To run a command in the background and redirect the standard output to a different file, enter:

   ```
   $ nohup find / -print >filenames &
   ```

   This example runs the **find / -print** command and stores its output in a file named `filenames`. Now only the process ID and prompt are displayed:

   ```
   677
   $
   ```

   Wait before logging off because the **nohup** command takes a moment to start the command specified by the *Command* parameter. If you log off too quickly, the command specified by the *Command* parameter may not run at all. Once the command specified by the *Command* parameter starts, logging off does not affect it.

3. To run more than one command, use a shell procedure. For example, if you write the shell procedure:

   ```
   neqn math1 | nroff > fmath1
   ```

   and name it the `nnfmath1` file, you can run the **nohup** command for all of the commands in the `nnfmath1` file with the command:

   ```
   nohup sh nnfmath1
   ```

4. If you assign execute permission to the `nnfmath1` file, you get the same results by issuing the command:

   ```
   nohup nnfmath1
   ```

5. To run the `nnfmath1` file in the background, enter:

   ```
   nohup nnfmath1  &
   ```

6. To run the `nnfmath1` file in the Korn shell, enter:

   ```
   nohup ksh nnfmath1
   ```

Related Information

The **csh** command, **nice** command, **sh** command.

The **signal** subroutine.

---

notifyevent Command

Purpose

Mails event information generated by the event response resource manager (ERRM) to a specified user ID.

Syntax

notifyevent [**-h**] [*user-ID*]

Description

The **enotifyevent** script always return messages in English. The language in which the messages of the **notifyevent** script are returned depend on the locale settings.

These scripts capture event information that is posted by the event response resource manager (ERRM) in environment variables that are generated by the ERRM when an event occurs. These scripts can be used as actions that are run by an event response resource. They can also be used as templates to create other user-defined actions.

See the *RSCT Guide and Reference* to understand how an event response resource runs an action command.

Event information is returned about the ERRM environment variables that are listed and described in the *RSCT Guide and Reference*, and also includes the following:

Local Time
: Time when the event or rearm event is observed. The actual environment variable supplied by ERRM is ERRM_TIME. This value is localized and converted to readable form before being displayed.

These scripts use the **mail** command to send event information to the specified user ID. When a user ID is specified, it is assumed to be valid, and it is used without verifying it. If a user ID is not specified, the user who is running the command is used as the default.

user-ID is the optional ID of the user to whom the event information will be mailed. If *user-ID* is not specified, the user who is running the command is used as the default.

Flags

–h Writes the script's usage statement to standard output.

Parameters

log_file

Specifies the name of the file where event information is logged. An absolute path for the *log_file* parameter should be specified.

The *log_file* is treated as a circular log and has a fixed size of 64KB. When *log_file* is full, new entries are written over the oldest existing entries.

If *log_file* already exists, event information is appended to it. If *log_file* does not exist, it is created so that event information can be written to it.

Exit Status

0 Command has run successfully.

Restrictions

1. These scripts must be run on the node where the ERRM is running.
2. The **mail** command is used to read the file.

Standard Output

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

Examples

1. Specify **user1** in Web-based System Manager to send mail to a user. The event response resource manager then runs the following command:

   ```
   /usr/sbin/rsct/bin/notifyevent user1
   ```

2. You can use the **mail** command to read the contents of the event information. The following example shows how a warning event for the **/var** file system (a file system resource) is formatted and logged:

   ```
   ============================================================================
   Event reported at Sun Mar 26 16:38:03 2002

   Condition Name:      /var space used
   Severity:     Warning
   Event Type:      Event
   Expression:      PercentTotUsed>90

   Resource Name:     /var
   Resource Class Name:    IBM.FileSystem
   Data Type:     CT_UINT32
   Data Value:     91
   ```

Location

/usr/sbin/rsct/bin/enotifyevent Contains the **enotifyevent** script

/usr/sbin/rsct/bin/notifyevent Contains the **notifyevent** script

Related Information

Commands: **mail**

Books: see the *RSCT Guide and Reference* for information about the event response resource manager (ERRM) and about how to use ERRM environment variables

nrglbd Daemon

Purpose
Manages the global location broker database.

Syntax
nrglbd [**-version**]

Description
The **glbd** daemon manages the global location broker (GLB) database. The GLB database, part of the Network Computing System (NCS), helps clients to clients to locate servers on a network or internet. The GLB database stores the locations (that is, the network addresses and port numbers) of servers on which processes are running. The **glbd** daemon maintains this database and provides access to it.

There are two versions of the GLB daemon, **glbd** and **nrglbd**. You should run only one **nrglbd** on a network or internet, and you should not run a **nrglbd** and a **glbd** on the same network or internet.

The **nrglbd** daemon is typically started in the background; it can be started in one of two ways:

- By a person with root user authority entering on the command line:
  ```
  /etc/ncs/nrglbd &
  ```
- Through the System Resource Controller (SRC), by entering on the command line:
  ```
  startsrc -s nrglbd
  ```

TCP/IP must be configured and running on your system before starting the **nrglbd** daemon. The **llbd** daemon must also be started and running before you start the **nrglbd** daemon.

Flags
-version Displays the version of NCS that this **nrglbd** belongs to, but does not start the daemon.

Files
/etc/rc.ncs Contains commands to start the NCS daemons.

Related Information
The **lb_admin** command.

The **llbd** daemon.

The Location Broker in *AIX 5L Version 5.2 Communications Programming Concepts*.

nroff Command

Purpose
Formats text for printing on typewriter-like devices and line printers.

Syntax
nroff [**-e**] [**-h**] [**-i**] [**-q**] [**-z**] [**-o** *List*] [**-n** *Number*] [**-s** *Number*] [**-r** *ANumber*] [**-u** *Number*] [**-T** *Name*] [**-man**] [**-me**] [**-mm**] [**-mptx**] [**-ms**] [*File* ... | **-**]

Description

The **nroff** command reads one or more files for printing on typewriter-like devices and line printers. If no file is specified or the **-** (minus sign) flag is specified as the last parameter, standard input is read by default. The *File* variable specifies files to be printed on a typewriter-like device by the **nroff** command. The default is standard input.

The **col** command may be required to postprocess **nroff** command output in certain cases.

Flags

-e Produces equally spaced words in adjusted lines, using the full resolution of a particular terminal.

-h Uses output tabs during horizontal spacing to speed output and reduce the output character count. Tab settings are assumed to be every eight nominal character widths.

-i Reads standard input after reading all specified files.

-man Selects the **man** macro processing package.

-me Selects the **me** macro processing package.

-mm Selects the **mm** macro processing package.

-mptx Selects the **mptx** macro processing package.

-ms Selects the **ms** macro processing package.

-n *Number* Assigns the specified number to the first printed page.

-o *List* Prints only those pages specified by the *List* variable, which consists of a comma-separated list of page numbers and ranges, as follows:

- A range of *Start-Stop* means print pages *Start* through *Stop.* For example, 9-15 prints pages 9 through 15.
- An initial *-Stop* means print from the beginning to page *Stop.*
- A final *Start-* means print from page *Start* to the end.
- A combination of page numbers and ranges prints the specified pages. For example, -3, 6-8,10,12- prints the beginning through page 3, pages 6 through 8, page 10, and page 12 to the end.

> **Note:** When the **-o***List* flag is used in a pipeline (as with one or more of the **eqn** or **tbl** commands) you may receive a `broken pipe` message if the last page in the document is not specified in the *List* parameter. This broken pipe message is not an indication of any problem and can be ignored.

-q Calls the simultaneous input/output mode of the **.rd** request.

-r *ANumber* Sets register *A* to the specified number. The value specified by the *A* variable must have a one-character ASCII name.

-s *Number* Stops every specified number of pages (the default is 1). The **nroff** command halts every specified number of pages to allow paper loading or changing, then resumes upon receipt of a linefeed or new-line character. This flag does not work in pipelines (for example, with the **mm** command). When the **nroff** command halts between pages, an ASCII BEL character is sent to the workstation.

-T *Name*

Prepares the output for the specified printing device. Typewriter-like devices and line printers use the following *Name* variables for international extended character sets, as well as English-language character sets, digits, and symbols:

hplj Hewlett-Packard LaserJet II and other models in the same series of printers.

ibm3812
3812 Pageprinter II.

ibm3816
3816 Pageprinter.

ibm4019
4019 LaserPrinter.

Note: The 4019 and the HP Laser Jet II printer both have nonprintable areas at the top and bottom of a page. If a file is targeted for these printers, be sure to define top and bottom margins (for example, by formatting with the **-mm** flag) so that all output can be positioned within the printable page.

37 Teletype Model 37 terminal (default) for terminal viewing only. This device does not support extended characters that are inputted by the \[N] form. Inputting Extended Single-Byte Characters provides more information.

lp Generic name for printers that can underline and tab. All text sent to the **lp** value using reverse linefeeds (for example, text that includes tables) must be processed with the **col** command. This device does not support extended characters that are inputted by the \[N] form. Inputting Extended Single-Byte Characters provides more information.

ppds Generic name for printers that support the personal printer data streams such as the Quietwriter III, Quickwriter, and Proprinters.

ibm5575
5575 Kanji Printer.

ibm5577
5577 Kanji Printer.

Note: For completeness of the text formatting system, the following devices are shipped *as is* from the AT&T Distribution center. No support is provided for these tables.

-T *Name* (Continued)

2631	Hewlett-Packard 2631 printer in regular mode.
2631-c	Hewlett-Packard 2631 printer in compressed mode.
2631-e	Hewlett-Packard 2631 printer in expanded mode.
300	DASI-300 printer.
300-12	DASI-300 terminal set to 12 characters per inch.
382	DTC-382.
4000a	Trendata 4000a terninal (4000A).
450	DASI-450 (Diablo Hyterm) printer.
450-12	DASI-450 terminal set to 12 characters per inch.
832	Anderson Jacobson 832 terminal.
8510	C.ITOH printer.
tn300	GE Terminet 300 terminal.
X	Printers equipped with a TX print train.
300s	DASI-300s printer (300S).
300s-12	DASI-300s printer set to 12 characters per inch (300S-12).

-u *Number* Sets the bold factor (number of character overstrokes) for the third font position (bold) to the specified number, or to 0 if the *Number* variable is missing.

-z Prints only messages generated by **.tm** (workstation message) requests.

> **Note:** See the article Macro Packages for Formatting Tools in the **troff** command for more inforamtion on the macros.

- Forces input to be read from standard input.

Files

/usr/share/lib/tmac/tmac.*	Contains pointers to standard macro files.
/usr/share/lib/macros/*	Contains standard macro files.
/usr/share/lib/nterm/*	Contains the terminal driving tables for the **nroff** command.
/usr/share/lib/pub/terminals	Contains a list of supported terminals.

Related Information

The **col** command, **mm** command, **neqn** command, **tbl** command, **troff** command.

The **nroff** and **troff** Input file format.

The article ″nroff and troff Requests for the nroff and troff Commands″ in the **troff** command.

---

nslookup Command

Purpose

Queries Internet domain name servers.

Syntax

nslookup [**-***Option* ...] [*Host*] [**-***NameServer*]

Description

The **nslookup** command queries Internet domain name servers in two modes. Interactive mode allows you to query name servers for information about various hosts and domains, or to print a list of the hosts in a domain. In noninteractive mode, the names and requested information are printed for a specified host or domain.

The **nslookup** command enters interactive mode when no arguments are given, or when the first argument is a - (minus sign) and the second argument is the host name or Internet address of a name server. When no arguments are given, the command queries the default name server. The - (minus sign) invokes an optional subcommand (*-Option...* variable). With the exception of the **set** command, these subcommands are specified on the command line and must precede the **nslookup** command arguments. The **set** subcommand options can alternatively be specified in the **.nslookuprc** file in the user's home directory.

The **nslookup** command executes in noninteractive mode when the first argument is the name or Internet address of the host being searched for. In this case, the host name or Internet address of the name server is optional.

The noninteractive command looks up information for the specified *Host* using the default name server or the name server specified by the *NameServer* parameter. If the *Host* parameter specifies an Internet address and the query type is **A** or **PTR**, the name of the host is returned. If the *Host* parameter specifies a name and the name does not have a trailing period, the default domain name is appended to the name. To look up a host not in the current domain, append a single period to the name.

> **Note:** If they are specified in the **.nslookuprc** file of the user's home directory, the **set** subcommand's **domain**, **srchlist**, **defname**, and **search** options can affect the behavior of the noninteractive command.

Subcommands

The following commands can be interrupted at any time by entering a Ctrl-C key sequence. To exit, enter a Ctrl-D key sequence or type `exit`. To treat a built-in command as a host name, precede it with an escape character, which is a \. Unrecognized commands are interpreted as host names.

The following subcommands are recognized by the **nslookup** command:

finger [*Name*] [**>** *FileName*]
finger [*Name*] [**>>** *FileName*]
: Connects with the **finger** daemon server on the current host. The current host is defined when a previous lookup for a host was successful and returned address information, such as that returned with the **set querytype=A** command. The *Name* parameter, which specifies a user name, is optional. The > and >> characters can be used to redirect output to a new or existing file.

server *Domain*
lserver *Domain*
: Changes the default server to the value specified by the *Domain* parameter. The **lserver** subcommand uses the initial server to look up information about the domain. The **server** subcommand uses the current default server. If an authoritative answer cannot be found, the names of any additional servers that might have the answer are returned.

root
: Changes the default server to the server for the root of the domain name space. Currently, the host `ns.nic.ddn.mil` is used. The name of the root server can be changed with the **set root** subcommand. (The **root** subcommand is synonymous with the **lserver ns.nic.ddn.mil** subcommand.)

ls [*Option*] *Domain* [**>** *FileName*]
ls [*Option*] *Domain* [**>>** *FileName*]

Lists the information available for the *Domain* specified, optionally creating or appending the output to the file specified by the *FileName* parameter. The default output contains host names and their Internet addresses. Possible values for the *Option* parameter are:

-t *QueryType*

Lists all records of the specified type. The default record type is **A**. Valid types are:

A	Host's Internet address
CNAME	Canonical name for an alias
HINFO	Host CPU and operating system type
KEY	Security Key Record
MINFO	Mailbox or mail list information
MX	Mail exchanger
NS	Nameserver for the named zone
PTR	Host name if the query is an Internet address; otherwise, the pointer to other information
SIG	Signature Record
SOA	Domain's ″start-of-authority″ information
TXT	Text information
UINFO	User information
WKS	Supported well-known services

-a Lists aliases of hosts in the domain (synonymous with the **-t CNAME** option).

-d Lists all records for the domain (synonymous with the **-t ANY** option).

-h Lists CPU and operating system information for the domain (synonymous with the **-t HINFO** option).

-s Lists well-known services of hosts in the domain (synonymous with the **-t WKS** option).

Note: When output is redirected to a file, hash marks are printed for every 50 records received from the server.

view *FileName*
Sorts the output of previous **ls** commands and lists them using the **more** command.

help
?
Prints a brief summary of commands.

exit
Exits the program.

set *Keyword*[=*Value*]

Changes state information that affects lookups. This subcommand can be specified on the command line or optionally included in the **.nslookuprc** file in the user's home directory. Valid keywords are:

all Prints the current values of the frequently used options to set. Information about the current default server and host is also printed.

class=*Value*
Changes the query class to one of the following. The class specifies the protocol group of the information. The default is **IN**.

IN Internet class

CHAOS
Chaos class

HESIOD
MIT Althena Hesiod class

ANY Wildcard (any of the above)

[no]debug
Turns debugging mode on. The default is **nodebug** (off).

[no]d2 Turns comprehensive debugging on. The default is **nod2** (off).

domain=*Name*
Changes the default domain name to the name specified by the *Name* parameter. The default domain name is appended to a lookup request, depending on the state of the **defname** and **search** options. The domain search list contains the parents of the default domain if the search list has at least two components in its name. For example, if the default domain is `CC.Berkeley.EDU`, the search list is `CC.Berkeley.EDU` and `Berkeley.EDU`. Use the **set srchlist** command to specify a different list. Use the **set all** command to display the list. The default of the **domain=***Name* option is the value specified in the system's **hostname**, **/etc/resolv.conf**, or **LOCALDOMAIN** file.

srchlst=*Name1***/***Name2***/...**
Changes the default domain name to the name specified by the *Name1* parameter, and changes the domain search list to the names specified by the *Name1*, *Name2*,..., parameters. A maximum of six names separated by slashes can be specified. Use the **set all** command to display the list of names. The default values are specified in the system's **hostname**, **/etc/resolv.conf**, or **LOCALDOMAIN** file.

> **Note:** This command overrides the default domain name and search list of the **set domain** command option.

[no]defname
Appends the default domain name to a single component lookup request (one that does not include a period). The default is **defname** (append).

[no]search
Appends the domain names in the domain search list to the request until an answer is received, if the lookup request contains a period other than a trailing period. The default is **search**.

port=*Value*
Changes the default TCP/UDP nameserver port to the number specified by the *Value* parameter. The default value is 53.

querytype=*Value*

type=*Value* Changes the information query to one of the following values. The default is **A**.

A Host's Internet address

ANY Any of the options available.

CNAME
Canonical name for an alias

HINFO Host CPU and operating system type

KEY Security Key Record

MINFO Mailbox or mail list information

MX Mail exchanger

NS Name server for the named zone

PTR Host name if the query is an Internet address; otherwise, the pointer to other information

SIG Signature Record

SOA Domain's ″start-of-authority″ information

TXT Text information

UINFO User information

WKS Supported well-known services

[no]recurse
Tells the name server to query other servers if it does not have information. The default is **recurse**.

retry=*Number*
Sets the number of times a request is retried to the value specified by the *Number* parameter. When a reply to a request is not received within the time frame specified by the **set timeout** command, the timeout period is doubled and the request resent. This subcommand controls the number of times a request is sent before timing out. The default value is 4.

root=*Host*
Changes the name of the root server to the name specified by the *Host* parameter. The default is `ns.nic.ddn.mil`.

timeout=*Number*
Changes the initial time-out interval for waiting for a reply to the number of seconds specified by the *Number* parameter. The default value is 5 seconds.

[no]vc Uses a virtual circuit when sending requests to the server. The default is **novc** (no virtual circuit).

[no]ignoretc
Ignores packet truncation errors. The default is **noignoretc** (do not ignore).

Examples

1. To change the default query type to host information (HINFO) and the initial time-out to 10 seconds, enter:

   ```
   nslookup -query=hinfo -timeout=10
   ```

2. To set the domain and the search list to three names, `lcs.MIT.EDU`, `ai.MIT.EDU`, and `MIT.EDU`, enter:

   ```
   nslookup -set srchlist=lcs.MIT.EDU/ai.MIT.EDU/MIT.EDU
   ```

This command overrides the default domain name and search list of the **set domain** command. Use the **set all** command to display the list.

3. To determine whether a name specifies a host, domain, or other entity, enter:

```
nslookup -querytype=ANY austin.ibm.com
```

The **nslookup** command returns all available information about the name `austin.ibm.com`, including Statement of Authority (SOA), name server, mail exchanger, and host Internet address information, as follows:

```
Server: benames.austin.ibm.com
Address: 9.3.199.2

austin.ibm.com origin = ausname1.austin.ibm.com
        mail addr = brian.chriss.austin.ibm.com
        serial=1993081210,refresh=3600,retry=300,expire=604800, min=86400
austin.ibm.com  nameserver = ausname1.austin.ibm.com
austin.ibm.com  nameserver = bb3names.austin.ibm.com
austin.ibm.com  nameserver = benames.austin.ibm.com
austin.ibm.com  nameserver = b45names.austin.ibm.com
austin.ibm.com  nameserver = bbcnames.austin.ibm.com
austin.ibm.com  nameserver = netmail.austin.ibm.com
austin.ibm.com  preference = 10, mail exchanger = netmail.austin.ibm.com
austin.ibm.com  inet address = 129.35.208.98
ausname1.austin.ibm.com  inet address = 129.35.17.2
bb3names.austin.ibm.com  inet address = 129.35.208.99
benames.austin.ibm.com   inet address = 9.3.199.2
b45names.austin.ibm.com  inet address = 129.35.49.2
bbcnames.austin.ibm.com  inet address = 129.35.17.68
netmail.austin.ibm.com   inet address = 129.35.208.98
```

4. To perform a noninteractive query on host `opus`, enter:

```
nslookup opus
```

The **nslookup** command responds similarly to the **host** command. The command returns the domain name and Internet address of host `opus`, as follows:

```
Name: opus.austin.ibm.com
Address: 129.35.129.223
```

If host `opus` had been a name server (a host running the **named** daemon with an empty **/etc/resolv.conf** file), the following information would have been displayed:

```
Server: loopback
Address: 0.0.0.0
```

Exit Status

When a lookup request is not successful, the **nslookup** command returns one of the following error messages:

Timed Out	Indicates the server did not respond to the request after the specified number of retries.
No Response from Server	Indicates that a name server is not running on the server machine.
No Records	Indicates the server does not have the resource records of the specified query type for the host, although the host name is valid.
Non-Existent Domain	Indicates the host or domain name does not exist.
Connection Refused	Indicates the connection to the name or finger server could not be made at the time of the inquiry. This error is typically associated with **ls** and **finger** requests.
Network Is Unreachable	Indicates the connection to the name or finger server could not be made at the time of the inquiry. This error is typically associated with **ls** and **finger** requests.

Server Failure	Indicates the name server encountered an internal inconsistency and could not return a valid answer.
Refused	Indicates the name server refused to service the request.
Format Error	Indicates the name server refused the request packet because it was not in the proper format.

Files

/usr/bin/nslookup	Contains the **nslookup** command.
/etc/resolv.conf	Contains the initial domain name and nameserver addresses.
$HOME/.nslookuprc	Contains the user's initial options.
HOSTALIASES	Contains the host aliases.
LOCALDOMAIN	Contains the override default domain.

Related Information

namerslv command, **traceroute** command.

named daemon.

res_query subroutine, **res_search** subroutine.

resolv.conf file format for TCP/IP.

TCP/IP Name Resolution in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

---

nsupdate Command

Purpose

Updates a DNS server.

Syntax

nsupdate [**-a**] [**-g**] [**-i**] [**-q**] [**-v**] [**-?**] [**-k** *KeyFile*] [**-h** *HostName*] [**-d** *DomainName*] [**-p** *PrimaryName*] [**-r** *IPAddress*] [**-s** *"CommandString"*]

Description

The **nsupdate** command updates the DNS server. The **nsupdate** command runs in either interactive mode or command mode. If a command string is provided, the **nsupdate** command runs the command string and then exits. The return code is dependent upon the successfulness of the command string.

The valid internal commands for the command string or interactive modes are:

r	Reset update packets. This must be first.
d	Delete a record. Following this command are questions for a record type and the value to delete.
a	Add a record. Following this command are questions for a record type and the value to add.
n	Add a record only if it doesn't exist yet. Following this command are questions for a record type and the value to add.
e	Add a record only if it already exists. Following this command are questions for a record type and the value to add.
t	Sets the default time to live value for the updated records.
s	Signs the update. Depending on if the **-a** or **-g** flags were specified, a key will be generated and the update will be signed.

x Transmit the update packet to the server specified by the **-p** flag.
v Turns on or off verbose mode.
i Returns the information passed in by the parameters.
p Prints the update packet in record format.
q Exits the command

The **-g** flag allows you to generate a set of keys to distribute to clients for use in secure mode. This flag takes the hostname and the primaryname and generates a public and a private key. For secure mode zone operation, the public is entered into the DNS server's database for the data to secure and the private key is placed on the client so that it can update that information at a later time.

The **-a** flag allows you to enter administrative mode. The zone may be secured by a zone key. This key gives the user full access to the zone. The **-a** flag tries to use the zone key for update signatures instead of the individual records key.

Flags

-a	Administrative mode. Attempts to use zone key instead of individual records key.
-d *DomainName*	Specifies the name of the domain to apply the update to. This is used with all records except PTR records.
-g	Generation mode. Used to generate a key pair for a primary name and a hostname.
-h *HostName*	Specifies the name of the record to update. This is used with all records except PTR records.
-i	Ignores errors and runs all the commands in the string.
-k *KeyFile*	Specifies the name of the default keyfile. This is the file for keys.
-p *PrimaryName*	Specifies the name or IP address of a DNS server. The primary DNS server is prefered.
-q	Turns off output.
-r *IPAddress*	Specifies the IP Address of the record to update. This is used only with PTR records.
-s *"CommandString"*	A set of internal commands separated by spaces or colons.
-v	Turns on verbose output.
-?	Command line options list

Exit Status

This command returns the following exit values:

0 Successful completion.
>0 An error occurred.

Security

Access Control: Any User

Example

To initialize a packet, delete all `A` records for the specified hostname, add an `A` record for the hostname to `9.3.145.2` association, signed and valid for 300 seconds with a default KEY pad of 3110400, transmit the packet, and quit, enter: (where ″;″ is pressing the enter key)

```
r;d;a;*;a;a;9.3.145.2;s;300;3110400;x;q
```

If any one of the items had failed, a message would be printed. In command line mode, an error would cause the program to exit and return 1.

Files

/usr/sbin/nsupdate	Contains the **nsupdate** command.
/usr/sbin/named	Contains the DNS server.

Related Information

DHCP Client Configuration File

DHCP Server Configuration File

bootp Configuration File

TCP/IP Address and Parameter Assignment - Dynamic Host Configuration Protocol (DHCP) in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

TCP/IP Daemons in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

---

nsupdate8 Command

Purpose

Generates a DNS update packet readable by a BIND 8 nameserver.

Syntax

nsupdate8 [**-v**] [**-d**] [*Filename*]

Description

The **nsupdate8** command can read from a file specified on the command line, from stdin for pipes or redirected input from a file, or interactively from a tty. All three methods use the same format specified below. The input defines a DNS update packet that can be used to update a ZONE. There are two sections to an update, a prerequisite section and an update section. The DNS name server verifies that all the prerequisites are true before processing the update section.

Flags

-d	Causes **nsupdate8** to generate additional debug information about its actions.
-v	Tells **nsupdate8** to use a virtual circuit (TCP connection), instead of the usual UDP connection.

The input format is defined as a set of update packets. Each packet is a set of strings terminated with a newline. The last string in the input stream may end with an EOF. If the stream is to contain multiple update packets, each packet must be separated from the next packet by a blank line (single newline character). The semi-colon is used a comment character. Anything after it is ignored and thrown out of the update packet.

The input format for nsupdate8 is a follows:

section opcode name [*ttl*] [*class*] [*type*] [*data*]

This is the general form. Each value of *section* and *opcode* modify what is required for later arguments.

section Defines the section of the update this record is for. Values are:

prereq Indicates the record is for the prerequisites section.

update Indicates the record is for the update section.

opcode Defines the operation to do with this record.

Values are:

Prerequisite operations:

nxdomain
Indicates that the name should be checked for non-existance. The ttl must be a non-zero value to indicate for how long it shouldn't exist. An optional class can be specified to restrict the search to only that class. The type of T_ANY is used as a wildcard to match any record type.

nydomain
Indicates that the name should be checked for existance. The ttl must be a non-zero value to indicate for how long the name should continue to exist. An optional class is allowed to restrict the search to only that class. The record type is T_NONE. This forces the check to make sure the name exists.

nxrrset Indicates that the record of a specific type doesn't exist for the name. An optional class and ttl are allowed to restrict the search. A type is mandatory.

nyrrset Indicates that the record of a specific type must exist for the name. The ttl and class are optional to restrict the search. The type and data are mandatory. Data may be a wild card. If the data is not a wildcard, it must match the format for the type specified.

Values are:

Update operations:

add Indicates that the record should be added to the zone. The type and data are mandatory. Wildcards are not allowed as data. The ttl is mandatory and must be non-zero. The class is optional.

delete Indicates that the record should be deleted from the zone. The type and data are optional. A wildcard is allowed for data. data defaults to the NULL string and type defaults to T_ANY. ttl and class are optional. If ttl is specified, it is reset to 0.

name The name of the DNS entry that one is testing or modifying.

[*ttl*] Optional time-to-live for the record being added. In some forms, this is not optional.

[*class*] Class of the record to be added to the zone. Values are IN, HESIOD, and CHAOS. The default for all messages is IN.

[*type*] The type of the record to be added to or checked against the zone. Values are A, NS, CNAME, SOA, MB, MR, NULL, WKS, PTR, HINFO, MINFO, MX, TXT, RP, AFSDB, X25, ISDN, RT, NSAP, NSAP_PTR, PX, and LOC. NOTE: The CNAME type may only be added with TSIG and TKEY records which are not currently supported in BIND 8.

[*data*] The data to be added or checked against the zone. The data should be valid for the specified type and in the DOMAIN data file format of a DNS server zone file. For prerequisite checking, an asterik (*) is used to match any value. This can also be used to delete all records of a particular type.

Here are the specific format cases:

```
prereq nxdomain <name> <ttl != 0> [class]
prereq nydomain <name ttl != 0> [class]
prereq nxrrset <name> [ttl] [class] <type>
prereq nyrrset <name> [ttl] [class] <type> <data>
update delete <name> [ttl] [class] [type] [data]
update add <name> <ttl != 0> [class] <type> <data>
```

Diagnostics

Messages indicating the different actions done and/or problems encountered by the program.

Related Information

The **nsupdate** command, **named** command,.

The **named.conf** file format, **DOMAIN Cache** file format, **DOMAIN Data** file format, **DOMAIN Reverse Data** file format, **DOMAIN Local Data** file format, **resolv.conf** file format.

TCP/IP Name Resolution and TCP/IP Daemons in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

Configuring Name Servers and Planning for DOMAIN Name Resolution in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

---

nsupdate9 Command

Purpose

Dynamic DNS update utility.

Syntax

nsupdate9 [**-d**] [**-y** keyname:*secret* | **-k** *keyfile*] [**-v**] [*filename*]

Description

The **nsupdate9** command is used to submit Dynamic DNS Update requests as defined in RFC2136 to a name server. This allows resource records to be added or removed from a zone without manually editing the zone file. A single update request can contain requests to add or remove more than one resource record.

Zones that are under dynamic control via **nsupdate9** or a DHCP server should not be edited by hand. Manual edits could conflict with dynamic updates and cause data to be lost.

The resource records that are dynamically added or removed with **nsupdate9** have to be in the same zone. Requests are sent to the zone's master server. This is identified by the MNAME field of the zone's SOA record.

The **-d** option makes **nsupdate9** operate in debug mode. This provides tracing information about the update requests that are made and the replies received from the name server.

Transaction signatures can be used to authenticate the Dynamic DNS updates. These use the TSIG resource record type described in RFC2845. The signatures rely on a shared secret that should only be known to **nsupdate9** and the name server. Currently, the only supported encryption algorithm for TSIG is HMAC-MD5, which is defined in RFC 2104. Once other algorithms are defined for TSIG, applications will need to ensure they select the appropriate algorithm as well as the key when authenticating each other. For instance suitable key and server statements would be added to **/etc/named.conf** so that the name server can associate the appropriate secret key and algorithm with the IP address of the client application that will be using TSIG authentication. **nsupdate9** does not read **/etc/named.conf.**

nsupdate9 uses the **-y** or **-k** option to provide the shared secret needed to generate a TSIG record for authenticating Dynamic DNS update requests. These options are mutually exclusive. With the **-k** option, **nsupdate9** reads the shared secret from the file *keyfile*, whose name is of the form **K{name}.+157.+{random}.private**. For historical reasons, the file **K{name}.+157.+{random}.key** must also be present. When the **-y** option is used, a signature is generated from keyname:*secret*. keyname is

the name of the key, and *secret* is the base64 encoded shared secret. Use of the **-y** option is discouraged because the shared *secret* is supplied as a command line argument in clear text. This may be visible in the output from ps(1) or in a history file maintained by the user's shell.

By default **nsupdate9** uses UDP to send update requests to the name server. The **-v** option makes **nsupdate9** use a TCP connection. This may be preferable when a batch of update requests is made.

Flags

-d Makes **nsupdate9** operate in debug mode.
-y keyname:*secret* Generates a signature from keyname:*secret*
-k *keyfile* Reads the shared secret from the file *keyfile*.
-v Makes **nsupdate9** use a TCP connection.

Parameters

filename File to be updated.

Input Format

nsupdate9 reads input from the file *filename* or standard input. Each command is supplied on exactly one line of input. Some commands are for administrative purposes. The others are either update instructions or prerequisite checks on the contents of the zone. These checks set conditions that some name or set of resource records (RRset) either exists or is absent from the zone. These conditions must be met if the entire update request is to succeed. Updates will be rejected if the tests for the prerequisite conditions fail.

Every update request consists of zero or more prerequisites and zero or more updates. This allows a suitably authenticated update request to proceed if some specified resource records are present or missing from the zone. A blank input line (or the send command) causes the accumulated commands to be sent as one Dynamic DNS update request to the name server.

The command formats and their meaning are as follows:

server [*servername*] [*port*]	Sends all dynamic update requests to the name server *servername*. When no **server** statement is provided, **nsupdate9** will send updates to the master server of the correct zone. The MNAME field of that zone's SOA record will identify the master server for that zone. *port* is the port number on *servername* where the dynamic update requests get sent. If no *port* number is specified, the default DNS port number of 53 is used.
local [*address*] [*port*]	Sends all dynamic update requests using the local address. When no local statement is provided, **nsupdate9** will send updates using an *address* and *port* choosen by the system. *port* can additionally be used to make requests come from a specific port. If no port number is specified, the system will assign one.
zone [*zonename*]	Specifies that all updates are to be made to the zone *zonename*. If no zone statement is provided, **nsupdate9** will attempt determine the correct zone to update based on the rest of the input.
key [*name*] [*secret*]	Specifies that all updates are to be TSIG signed using the *keyname keysecret* pair. The key command overrides any key specified on the command line via **-**y or **-k**.
prereq nxdomain [*domain-name*]	Requires that no resource record of any type exists with name *domain-name*.
prereq yxdomain [*domain-name*]	Requires that *domain-name* exists (has as at least one resource record, of any type).

prereq nxrrset [*domain-name*] [*class*] [*type*]	Requires that no resource record exists of the specified *type*, *class* and *domain-name*. If class is omitted, IN (internet) is assumed.
prereq yxrrset [*domain-name*] [*class*] [*type*]	This requires that a resource record of the specified *type*, *class* and *domain-name* must exist. If class is omitted, IN (internet) is assumed.
prereq yxrrset [*domain-name*] [*class*] [*type*] [*data...*]	The data from each set of prerequisites of this form sharing a common *type*, *class*, and *domain-name* are combined to form a set of RRs. This set of RRs must exactly match the set of RRs existing in the zone at the given *type*, *class*, and *domain-name*. The *data* are written in the standard text representation of the resource record's RDATA.
update delete [*domain-name*] [*ttl*] [*class*] [*type*] [*data...*]	Deletes any resource records named *domain-name*. If *type* and *data* is provided, only matching resource records will be removed. The internet *class* is assumed if class is not supplied. The *ttl* is ignored, and is only allowed for compatibility.
update add [*domain-name*] [*ttl*] [*class*] [*type*] [*data...*]	Adds a new resource record with the specified *ttl*, *class* and *data*.
show	Displays the current message, containing all of the prerequisites and updates specified since the last send.
send	Sends the current message. This is equivalent to entering a blank line.

Lines beginning with a semicolon are comments and are ignored.

Examples

The examples below show how **nsupdate9** could be used to insert and delete resource records from the example.com zone. Notice that the input in each example contains a trailing blank line so that a group of commands are sent as one dynamic update request to the master name server for example.com.

```
# nsupdate9
> update delete oldhost.example.com A
> update add newhost.example.com 86400 A 172.16.1.1
>
```

Any A records for oldhost.example.com are deleted. and an A record for newhost.example.com it IP address 172.16.1.1 is added. The newly-added record has a 1 day TTL (86400 seconds)

```
# nsupdate9
> prereq nxdomain nickname.example.com
> update add nickname.example.com CNAME somehost.example.com
>
```

The prerequisite condition gets the name server to check that there are no resource records of any type for **nickname.example.com**. If there are, the update request fails. If this name does not exist, a CNAME for it is added. This ensures that when the CNAME is added, it cannot conflict with the long-standing rule in RFC1034 that a name must not exist as any other record type if it exists as a CNAME. (The rule has been updated for DNSSEC in RFC2535 to allow CNAMEs to have SIG, KEY and NXT records.)

Files

/etc/resolv.conf	Used to identify default name server
K{name}.+157.+{random}.key	Base-64 encoding of HMAC-MD5 key created by **dnssec-keygen**(8).
K{name}.+157.+{random}.privat	Base-64 encoding of HMAC-MD5 key created by **dnssec-keygen**(8).

Related Information

RFC2136, RFC3007, RFC2104, RFC2845, RFC1034, and RFC2535.

The **named9** daemon and **dnssec-keygen** command.

ntpdate Command

Purpose

Sets the date and time using the Network Time Protocol (NTP). This command only applies to AIX 4.2 or later.

Syntax

ntpdate [**-b**] [**-d**] [**-s**] [**-u**] [**-a** *Keyid*] [**-e** *AuthenticationDelay*] [**-k** *KeyFile*] [**-o** *Version*] [**-p** *Samples*] [**-t** *TimeOut*] *Server* ...

Description

The **ntpdate** command sets the local date and time by polling the NTP servers specified to determine the correct time. It obtains a number of samples from each server specified and applies the standard NTP clock filter and selection algorithms to select the best of the samples.

The **ntpdate** command makes time adjustments in one of the following ways:

- If it determines that the clock is off by more than 0.5 seconds, it steps the clock's time by calling the **settimeofday** subroutine. This is the preferred method at boot time.
- If it determines that the clock is off by less than 0.5 seconds, it slews the clock's time by calling the **adjtime** subroutine with the offset. This method tends to keep a badly drifting clock more accurate, though at some expense to stability. When running the **ntpdate** command on a regular basis from the **cron** command instead of running a daemon, doing so once every hour or two results in precise enough timekeeping to avoid stepping the clock.

Notes:

1. The **ntpdate** command's reliability and precision improves dramatically with a greater number of servers. Although you can use a single server, you obtain better performance by providing at least three or four servers.
2. If an NTP server daemon like the **xntpd** daemon is running on the same host, the **ntpdate** command will decline to set the date.
3. You must have root authority on the local host to run this command.

Flags

-a *Keyid*	Authenticate all packets using *Keyid*.
-b	Step the clock's time by calling the **settimeofday** subroutine.
-d	Specifies debug mode. Determines what results the **ntpdate** command produces without actually doing them. The results appear on the screen. This flag uses unprivileged ports.
-e *AuthenticationDelay*	Specifies the amount of time in seconds to delay the authentication processing.
-k *KeyFile*	Specifies a different name for the file containing the keys when not using the default **/etc/ntp.keys** file. See ... for the description of the *KeyFile*.
-o *Version*	Specifies the NTP version implementation to use when polling its outgoing packets. The values for *Version* can be 1, 2 or 3. The default is 3.
-p *Samples*	Specifies the number of samples to acquire from each server. The values for *Samples* can be between 1 and 8 inclusive. The default is 4.
-s	Specifies the use of the syslog facility to log actions instead of using standard output. Useful when running the **ntpdate** command with the **cron** command.
-t *TimeOut*	Specifies the amount of time to wait for a response. The value given for *TimeOut* is rounded to a multiple of 0.2 seconds. The default is 1 second.

-u Specifies the use of an unprivileged port to send the packets from. Useful when you are behind a firewall that blocks incoming traffic to privileged ports, and you want to synchronize with hosts beyond the firewall. A firewall is a system or machine that controls the access from outside networks to a private network.

Parameters

Server ... Specifies the servers to poll.

Exit Status

This command returns the following exit values:

0 Successful completion.
>0 An error occurred.

Security

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

Auditing Events: N/A

Examples

To set the local date and time by polling the NTP servers at address `9.3.149.107`, enter:

```
/usr/sbin/ntpdate 9.3.149.107
```

Output similar to the following appears:

```
28 Feb 12:09:13 ntpdate [18450]: step time server 9.3.149.107
offset 38.417792 sec
```

Files

/usr/sbin/ntpdate Contains the **ntpdate** command.
/etc/ntp.keys Contains the default key file.

Related Information

Commands: **ntpq**, **ntptrace**, **xntpdc**

Daemons: **xntpd**,

---

ntpq Command

Purpose

Starts the standard Network Time Protocol (NTP) query program. This command only applies to AIX 4.2 or later.

Syntax

ntpq [**-i**] [**-n**] [**-p**] [**-c** *SubCommand*] [*Host* ...]

Description

The **ntpq** command queries the NTP servers running on the hosts specified which implement the recommended NTP mode 6 control message format about current state and can request changes in that state. It runs either in interactive mode or by using command-line arguments. You can make requests to read and write arbitrary variables, and raw and formatted output options are available. The **ntpq** command can also obtain and print a list of peers in a common format by sending multiple queries to the server.

If you enter the **ntpq** command with one or more flags, the NTP servers running on each of the hosts specified (or defaults to local host) receive each request. If you do not enter any flags, the **ntpq** command tries to read commands from standard input and run them on the NTP server running on the first host specified or on the local host by default. It prompts for subcommands if standard input is the terminal.

The **ntpq** command uses NTP mode 6 packets to communicate with the NTP server and can query any compatible server on the network which permits it.

The **ntpq** command makes one attempt to retransmit requests, and will time-out requests if the remote host does not respond within a suitable time.

Specifying a flag other than **-i** or **-n** sends the queries to the specified hosts immediately. Otherwise, the **ntpq** command attempts to read interactive format subcommands from standard input.

Flags

-c *SubCommand*	Specifies an interactive format command. This flag adds *SubCommand* to the list of commands to run on the specified hosts. You can enter multiple **-c** flags.
-i	Specifies interactive mode. Standard output displays prompts and standard input reads commands.
-n	Displays all host addresses in dotted decimal format (x.x.x.x) rather than the canonical host names.
-p	Displays a list of the peers known to the server and a summary of their state. Same as using the **peers** subcommand.

Parameters

Host ...	Specifies the hosts.

Exit Status

This command returns the following exit values:

0	Successful completion.
>0	An error occurred.

Security

Access Control: You must be part of the system group to run this command.

Auditing Events: N/A

Examples

1. To start the Network Time Protocol query program in interactive mode, type:

   ```
   ntpq -i
   ```

2. To add a time interval of 1000 milliseconds to timestamps, type:

```
ntpq -c "delay 1000"
```

ntpq Internal Subcommands

The following subcommands can only be used while running the **ntpq** query program.

Interactive Format Subcommands

Interactive format subcommands consist of a keyword followed by zero to four arguments. You only need to type enough characters of the full keyword to uniquely identify the subcommand. The output of a subcommand goes to standard output, but you can redirect the output of individual subcommands to a file by appending a > (greater than sign), followed by a file name, to the command line.

Some interactive format subcommands run entirely within the **ntpq** query program and do not result in sending NTP mode 6 requests to a server.

The data carried by NTP mode 6 messages consists of a list of items of the form:

Variable=Value

where *Value* is ignored, and can be omitted, in requests to the server to read variables. The **ntpq** query program maintains an internal list where data to be included in control messages can be assembled and sent using the **readlist** and **writelist** control message subcommands.

? [*SubCommand*]	Displays command usage information. When used without *SubCommand*, displays a list of all the **ntpq** command keywords. When used with *SubCommand*, displays function and usage information about the subcommand.
addvars *Variable* [=*Value*] [,...]	Specifies the variables and their optional values to be added to the internal data list. If adding more than one variable, the list must be separated by commas and not contain spaces.
authenticate yes | **no**	Specifies whether to send authentication with all requests or not. Normally the **ntpq** query program does not authenticate requests unless they are write requests.
clearvars	Removes all variables from the internal data list.
cooked	Displays all results received from the remote server reformatted. A trailing ? (question mark) marks variables that do not have decodeable values.
debug more | **less** | **off**	Turns the **ntpq** query program debugging on or off. The **more** and **less** options control the verbosity of the output. If you enter this subcommand without an argument, it prints the current setting for this subcommand.
delay *MilliSeconds*	Specifies the time interval to add to timestamps included in requests which require authentication. This subcommand enables unreliable server reconfiguration over long delay network paths or between machines whose clocks are unsynchronized. If you enter this subcommand without an argument, it prints the current setting for this subcommand.
host *HostName*	Specifies the host to send queries to. *HostName* may be either a host name or a numeric address. If you enter this subcommand without an argument, it prints the current setting for this subcommand.
hostnames yes | **no**	Specifies whether to output the host name (**yes**) or the numeric address (**no**). Defaults to **yes** unless the **-n** flag is used. If you enter this subcommand without an argument, it prints the current setting for this subcommand.
keyid *Number*	Specifies the server key number to use to authenticate configuration requests. If you enter this subcommand without an argument, it prints the current setting for this subcommand.
ntpversion 1 | **2** | **3**	Specifies the NTP version implementation to use when polling its packets. The default is 3. If you enter this subcommand without an argument, it prints the current setting for this subcommand. **Note:** Mode 6 control messages and modes did not exist in NTP version 1.
passwd	Prompts you to type in the NTP server authentication password to use to authenticate configuration requests.
quit	Exits the **ntpq** query program.

raw — Displays all results received from the remote server without formatting. Only transforms non-ascii characters into printable form.

rmvars *Variable* [=*Value*] [,...] — Specifies the variables and their optional values to be removed from the internal data list. If removing more than one variable, the list must be separated by commas and not contain spaces.

timeout *MilliSeconds* — Specifies the time-out period for responses to server queries. The default is 5000 milliseconds. If you enter this subcommand without an argument, it prints the current setting for this subcommand.
Note: Because **ntpq** query program retries each query once after a time-out, the total waiting time for a time-out is twice the time-out value set.

Control Message Subcommands

Each peer known to an NTP server has a 16-bit integer association identifier assigned to it. NTP control messages which carry peer variables must identify the peer that the values correspond to by including its association ID. An association ID of 0 is special and indicates the variables are system variables whose names are drawn from a separate name space.

The **ntpq** control message subcommands result in one or more NTP mode 6 messages sent to the server, and outputs the data returned in some format. Most subcommands currently implemented send a single message and expect a single response. The current exceptions are the **peers** subcommand, which sends a preprogrammed series of messages to obtain the data it needs, and the **mreadlist** and **mreadvar** subcommands, which iterate over a range of associations.

associations — Obtains and prints a list of association identifiers and peer statuses for in-spec peers of the server being queried. The list is printed in the following columns:

- First column contains the index numbering the associations from 1 for internal use.
- Second column contains the actual association identifier returned by the server.
- Third column contains the status word for the peer.
- Remaining columns contain data decoded from the status word.

Note: The data returned by the **associations** subcommand is cached internally in the **ntpq** query program. When dealing with servers that use difficult association identifiers, use the index as an argument, in the form `&index`, as an alternative to the association identifier.

clockvar [*AssocID*] [*Variable* [=*Value*], ...]
or
cv [*AssocID*] [*Variable* [=*Value*], ...] — Displays a list of the server's clock variables. Servers which have a radio clock or other external synchronization respond positively to this. To request the system clock variables, leave *AssocID* blank or type `0`. If the server treats clocks as pseudo-peers and can possibly have more than one clock connected at once, referencing the appropriate peer association ID shows the variables of a particular clock. Omitting the variable list causes the server to return a default variable display.

lassociations — Displays a list of association identifiers and peer statuses for all associations for which the server is maintaining state. This subcommand differs from the **associations** subcommand only for servers which retain state for out-of-spec client associations.

lpassociations — Displays data for all associations, including out-of-spec client associations, from the internally cached list of associations.

lpeers — Displays a summary of all associations the server maintains state for Similar to the **peers** subcommand. This may produce a longer list of peers from out-of-spec client servers.

mreadvar *AssocID AssocID* [*Variable* [=*Value*], ...]
or
mrv *AssocID AssocID* [*Variable* [=*Value*], ...] — Displays the values of the specified peer variables for each server in the range of given nonzero association IDs. The association list cached by the most recent associations command determines the range.

mreadlist *AssocID AssocID*
or
mrl *AssocID AssocID*

Displays the values of the specified peer variables in the internal variable list for each server in the range of given nonzero association IDs. The association list cached by the most recent associations command determines the range.

opeers

An old form of the **peers** subcommand. Replaces the reference ID with the local interface address.

passociations

Displays association data concerning in-spec peers from the internally cached list of associations. This subcommand works like the **associations** subcommand except that it displays the internally stored data rather than making a new query.

peers

Displays a list of in-spec peers of the server and a summary of each peer's state. Summary information includes the following:

- Address of the remote peer
- Reference ID (0.0.0.0 for an unknown reference ID)
- Stratum of the remote peer (a stratum of 16 indicates the remote peer is unsynchronized)
- Type of peer (local, unicast, multicast, *or* broadcast)
- Time the last packet was received, the polling interval (seconds)
- Polling interval (seconds)
- Reachability register (octal)
- Current estimated delay, offset and dispersion of the peer (seconds)

The character in the left margin indicates the fate of this peer in the clock selection process:

space Discarded due to high stratum and/or failed sanity checks.

x Designated falseticker by the intersection algorithm.

. Culled from the end of the candidate list.

- Discarded by the clustering algorithm.

+ Included in the final selection set.

Selected for synchronization but distance exceeds maximum.

***** Selected for synchronization.

o Selected for synchronization, **pps** signal in use.

The contents of the host field may be a host name, an IP address, a reference clock implementation name with its parameter or REFCLK (*ImplementationNumber*, *Parameter*). Only IP addresses display when using **hostnames no**.

Note:

The **peers** subcommand depends on the ability to parse the values in the responses it gets. It may fail to work from time to time with servers that poorly control the data formats.

The **peers** subcommand is non-atomic and may occasionally result in spurious error messages about invalid associations occurring and terminating the command.

pstatus *AssocID*

Displays the names and values of the peer variables of the server with the given association by sending a read status request. The output displays the header preceding the variables, both in hexidecimal and in English.

readlist [*AssocID*]
or
rl [*AssocID*]

Displays the values of the peer variables in the internal variable list of the server with the given association. To request the system variables, leave *AssocID* blank or type `0`. If the internal variable list is empty, the server returns a default variable display.

readvar [*AssocID*] [*Variable* [=*Value*], ...] or **rv** [*AssocID*] [*Variable* [=*Value*], ...]	Displays the values of the specified peer variables of the server with the given association by sending a read variables request. To request the system variables, leave *AssocID* blank or type `0`. Omitting the variable list causes the server to return a default variable display.
writevar [*AssocID*] [*Variable* [=*Value*], ...]	Writes the values of the specified peer variables to the server with the given association by sending a write variables request.
writelist [*AssocID*]	Writes the values of the peer variables in the internal variable list of the server with the given association.

Files

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

Related Information

Commands: **ntpdate**, **ntptrace**, **xntpdc**

Daemons: **xntpd**,

---

ntptrace Command

Purpose

Traces a chain of Network Time Protocol (NTP) hosts back to their master time source.

Syntax

ntptrace [**-d**] [**-n**] [**-v**] [**-r** *Retries*] [**-t** *TimeOut*] [*Server*]

Description

The **ntptrace** command determines where a given NTP server gets its time, and follows the chain of NTP servers back to their master time source. For example, stratum 0 server.

Flags

-d	Turns on debugging output.
-n	Outputs host IP addresses instead of host names.
-r *Retries*	Specifies the number of retransmission attempts for each host. The default is 5.
-t *TimeOut*	Specifies the retransmission timeout in seconds. The default is 2 seconds.
-v	Specifies verbose mode.

Parameters

Server	Specifies the server. The default is the local host.

Exit Status

This command returns the following exit values:

0	Successful completion.
>0	An error occurred.

Security

Access Control: You must be part of the system group to run this command.

Auditing Events: N/A

Examples

To trace where the local host NTP server gets its time from, enter:

```
ntptrace
```

Output similar to the following appears:

```
localhost: stratum 4, offset 0.0019529, synch distance 0.144135
server2.bozo.com: stratum 2, offset 0.0124263, synch distance 0.115784
usndh.edu: stratum 1, offset 0.0019298, synch distance 0.011993, refid
'WWVB'
```

On each line, the fields are:

1. the host's stratum,
2. the time offset between that host and the local host, as measured by the **ntptrace** command, (this is why it is not always zero for localhost).
3. the host's synchronization distance, which is a measure of the quality of the clock's time, and
4. the reference clock ID This only applies to stratum-1 servers.

All times are given in seconds.

Files

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

Related Information

Commands: **ntpq**, **ntpdate**, **xntpdc**

Daemons: **xntpd**,

---

ntsc Command

Purpose

Enables or disables NTSC (National Television Standards Committee) video output of the G10 Graphics with Motion Video Adapter.

Syntax

ntsc [**-v**] [on | off]

Description

The **ntsc** command enables or disables the NTSC video output of the G10 Graphics with Motion Video Adapter. When the NTSC video output is enabled, the same image displayed on LCD and/or CRT is displayed on NTSC video output. While video capture or playback application is running, the **ntsc** command can not enable NTSC video output. If on or off parameter is not specified, the **ntsc** command usage is displayed.

Flags

-v Displays the result of the action taken.

Parameters

on Enables NTSC video output, if a video capture or playback application is not running.
off Disables NTSC video output.

Security

Access Control: Any User

Auditing Events: N/A

Examples

1. To enable NTSC video output, enter:
   ```
   ntsc on
   ```
2. To disable NTSC video output with result message, enter:
   ```
   ntsc -v off
   ```

   Something similar to the following displays:
   ```
   ntsc off succeeded
   ```

Files

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

---

nulladm Command

Purpose

Creates active accounting data files.

Syntax

/usr/sbin/acct/nulladm [*File ...*]

Description

The **nulladm** command creates the file specified by the *File* parameter, gives read (r) and write (w) permission to the file owner, and group and read (r) permission to other users, and ensures that the file owner and group are **adm**. Various accounting shell procedures call the **nulladm** command. A user with administrative authority can use this command to set up the active data files, such as the **/var/adm/wtmp** file.

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

Security

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

Files

/usr/sbin/acct	Contains the accounting commands.
/var/adm/acct/sum	Contains accounting data files.

Related Information

The **acctmerg** command, **prdaily** command.

Accounting Overview in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*.

Setting Up an Accounting System in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*.

Monitoring and Tuning Commands and Subroutines in *AIX 5L Version 5.2 Performance Management Guide*.

---

number Command

Purpose

Displays the written form of a number.

Syntax

number

Description

The **number** command translates the numerical representation of an entered number to the written form. The largest number it can translate accurately contains 66 digits. For example:

```
12345678
twelve million.
three hundred forty five thousand.
six hundred seventy eight.
```

In the above example, you entered `12345678` and the computer translated it to `twelve million three hundred forty five thousand six hundred seventy eight`.

The **number** command does not prompt you for a number. Once started, it simply waits for input. To exit the program, press the Interrupt (Ctrl-C) or End Of File **(**Ctrl-D) key sequence.

Files

/usr/games	Contains the system games.

Related Information

The **arithmetic** command, **back** command, **bj** command, **craps** command, **fish** command, **fortune** command, **hangman** command, **moo** command, **quiz** command, **ttt** command, **turnoff** command, **turnon** command, **wump** command.

od Command

Purpose

Displays files in a specified format.

Syntax

To Display Files Using a Type-String to Format the Output

od [**-v**] [**-A** *AddressBase*] [**-N** *Count*] [**-j** *Skip*] [**-t** *TypeString* ...] [*File* ...]

To Display a File Using Flags to Format the Output

od [**-a**] [**-b**] [**-c**] [**-C**] [**-d**] [**-D**] [**-e**] [**-f**] [**-F**] [**-h**] [**-H**] [**-i**] [**-l**] [**-l**] [**-L**] [**-o**] [**-O**] [**-p**] [**-P**] [**-s**] [**-v**] [**-x**] [**-X**] [[**-S** [*N*]] [**-w** [*N*]] [*File*] [[**+**] *Offset* [**.** | **b** | **B**] [**+**] *Label* [**.** | **b** | **B**]]

Description

The **od** command displays the file specified by the *File* parameter in the format specified. If the *File* parameter is not given, the **od** command reads standard input. Multiple types can be specified by using multiple -bcCDdFfOoSstvXx options.

In the first syntax format, the output format is specified by the **-t** flag. If no format type is specified, **-t o2** is the default.

In the second syntax format, the output format is specified by a combination of flags. The *Offset* parameter specifies the point in the file where the file output begins. By default, the *Offset* parameter is interpreted as octal bytes. If the **.** (dot) suffix is appended, the parameter is interpreted as a decimal; if the parameter begins with a leading `x` or `0x`, it is treated as a hexadecimal. If the **b** suffix is added to the parameter, it is interpreted in blocks of 512 bytes; if the **B** suffix is added to the parameter, it is interpreted in blocks of 1024 bytes.

The *Label* parameter is interpreted as a pseudo-address for the first byte displayed. If used, it is given in **( )** (parentheses) following the *Offset* parameter. The suffixes have the same meanings as for the *Offset* parameter.

When the **od** command reads standard input, the *Offset* parameter and the *Label* parameter must be preceded by a **+** (plus sign).

The setting of environment variables such as **LANG** and **LC_ALL** affects the operation of the **od** command. See ″Understanding Locale Environment Variables″ in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices* for more information.

Flags

The flags for the first format are:

-A *AddressBase*

Specifies the input offset base. The *AddressBase* variable is one of the following characters:

d Offset base is written in decimal.

o Offset base is written in octal.

x Offset base is written in hexadecimal.

n Offset base is not displayed.

Unless **-A n** is specified, the output line will be preceded by the input offset, cumulative across input files, of the next byte to be written. In addition, the offset of the byte following the last byte written will be displayed after all the input data has been processed. Without the **-A** address_base option and the [offset_string] operand, the input offset base is displayed in octal.

-j *Skip*

Jumps over the number of bytes given by the *Skip* variable before beginning to display output. If more than one file is specified, the **od** command jumps over the designated number of bytes of the concatenated input files before displaying output. If the combined input is not at least the length of the skip bytes, the **od** command will write a diagnostic message to standard error and exit non-zero status.

By default, the value of the *Skip* variable is interpreted as a decimal number. With a leading 0x or 0X, the offset is interpreted as a hexadecimal number; otherwise, with a leading 0, the offset shall be interpreted as an octal number. If the characters **b**, **k**, or **m** are appended to the number contained by the *Skip* variable, the offset is equal to the value, in bytes, of the *Skip* variable multiplied by 512, 1024, or 1024*1024, respectively.

-N *Count*

Formats no more than the number of input bytes specified by the *Count* variable. By default, the value of the *Count* variable is interpreted as a decimal number. With a leading 0x or 0X, it is treated as a hexadecimal number. If it begins with a 0, it is treated as an octal number. The base of the address displayed is not implied by the base of the *Count* option-argument.

-t *TypeString*

Specifies the output type. The *TypeString* variable is a string specifying the types to be used when writing out data. Multiple types can be concatenated within the same *TypeString* variable, and the **-t** flag can be specified more than once. Output lines are written for each type specified, in the order in which the type specification characters are given. The *TypeString* variable can consist of the following characters:

a Displays bytes as named characters. Bytes with the least seven bits in the range of 0 through 01777 are written using the corresponding names for those characters.

c Displays bytes as characters. The number of bytes transformed by the **c** type string is determined by the **LC_CTYPE** local category. Printable multibyte characters are written in the area corresponding to the first byte of the character; the two character sequence ** is written in the area corresponding to each remaining byte in the character, as an indication that the character is continued. The following nongraphic characters are used as C-language escape sequences:

```
\    Backslash
\a   Alert
\b   Backspace
\f   Form-feed
\n   New-line character
\0   Null
\r   Carriage return
\t   Tab
\v   Vertical tab
```

d Displays bytes as signed decimals. By default, the **od** command transforms the corresponding number of bytes in the C-language type **int**. The **d** type string can be followed by an unsigned decimal integer that specifies the number of bytes to be transformed by each instance of the output type.

An optional **C**, **I**, **L**, or **S** character can be appended to the **d** option, indicating that the conversion should be applied to an item of type **char**, **int**, **long**, or **short**, respectively.

f Displays bytes as floating points. By default, the **od** command transforms the corresponding number of bytes in the C-language type **double**. The **f** type string can be followed by an unsigned decimal integer that specifies the number of bytes to be transformed by each instance of the output type.

An optional **F**, **D**, or **L** character can be appended to the **f** option, indicating that the conversion should be applied to an item of type **float**, **double**, or **long double**, respectively.

o Displays bytes as octals. By default, the **od** command transforms the corresponding number of bytes in the C-language type **int**. The **o** type string can be followed by an unsigned decimal integer that specifies the number of bytes to be transformed by each instance of the output type.

An optional **C**, **I**, **L**, or **S** character can be appended to the **o** option, indicating that the conversion should be applied to an item of type **char**, **int**, **long**, or **short**, respectively.

u Display bytes as unsigned decimal. By default, the **od** command transforms the corresponding number of bytes in the C-language type **int**. The **u** type string can be followed by an unsigned decimal integer that specifies the number of bytes to be transformed by each instance of the output type.

An optional **C**, **I**, **L**, or **S** character can be appended to the **u** option, indicating that the conversion should be applied to an item of type **char**, **int**, **long**, or **short**, respectively.

x Display bytes as hexadecimal. By default, the **od** command transforms the corresponding number of bytes in the C-language type **int**. The **x** type string can be followed by an unsigned decimal integer that specifies the number of bytes to be transformed by each instance of the output type.

An optional **C**, **I**, **L**, or **S** character can be appended to the **x** option, indicating that the conversion should be applied to an item of type **char**, **int**, **long**, or **short**, respectively.

The flags for the second format are:

-a Displays bytes as characters and displays them with their ASCII names. If the **-p** flag is also given, bytes with even parity are underlined. The **-P** flag causes bytes with odd parity to be underlined. Otherwise, parity is ignored.

-b Displays bytes as octal values.

-c Displays bytes as ASCII characters. The following nongraphic characters appear as C-language escape sequences:

```
\    Backslash
\a   Alert
\b   Backspace
\f   Form-feed
\n   New-line character
\0   Null
\r   Carriage return
\t   Tab
\v   Vertical tab
```

Others appear as three-digit octal numbers.

-C Displays extended characters as standard printable ASCII characters (using the appropriate character escape string) and displays multibyte characters in hexadecimal form.

-d Displays 16-bit words as unsigned decimal values.

-D Displays long words as unsigned decimal values.

-e Displays long words as double-precision, floating point. (same as the **-F** flag)

-f Displays long words as floating points.

-F Displays long words as double-precision, floating point. (same as the **-e** flag)

-h Displays 16-bit words as unsigned hexadecimal.

-H Displays long words as unsigned hexadecimal values.

-i Displays 16-bit words as signed decimal.

-I (Uppercase i) Displays long words as signed decimal values.

-l (Lowercase L) Displays long words as signed decimal values.

-L Displays long words as signed decimal values.

Note: The flags **-I** (uppercase i), **-l** (lowercase L), and **-L** are identical.

-o Displays 16-bit words as unsigned octal.

-O Displays long words as unsigned octal values.

-p Indicates even parity on **-a** conversion.

-P Indicates odd parity on **-a** conversion.

-s Displays 16-bit words as signed decimal values.

-S[*N*] Searches for strings of characters ending with a null byte. The *N* variable specifies the minimum length string to be recognized. If the *N* variable is omitted, the minimum length defaults to 3 characters.

The **-v** flag is the same for both formats:

-v Writes all input data. By default, output lines that are identical to the immediately preceding output lines are not printed, but are replaced with a line containing only an * (asterisk). When the **-v** flag is specified, all the lines are printed.

-w [*N*] Specifies the number of input bytes to be interpreted and displayed on each output line. If the **-w** flag is not specified, 16 bytes are read for each display line. If the **-w** flag is specified without the *N* variable, 32 bytes are read for each display line. The maximum input value is 4096 bytes. Input values greater than 4096 bytes will be reassigned the maximum value.

-x Displays 16-bit words as hexadecimal values.

-X Displays long words as unsigned hexadecimal values. (same as the **-H** flag)

Exit Status

This command returns the following exit values:

0 All input files were processed successfully.

>0 An error occurred.

Examples

1. To display a file in octal, a page at a time, enter:

   ```
   od a.out | pg
   ```

   This command displays the `a.out` file in octal format and pipes the output through the **pg** command.
2. To translate a file into several formats at once, enter:

   ```
   od -t cx a.out > a.xcd
   ```

   This command writes the contents of the `a.out` file, in hexadecimal format (**x**) and character format (**c**), into the `a.xcd` file.
3. To start displaying a file in the middle (using the first syntax format), enter:

   ```
   od -t acx -j 100 a.out
   ```

   This command displays the `a.out` file in named character (**a**), character (**c**), and hexadecimal (**x**) formats, starting from the 100th byte.
4. To start in the middle of a file (using the second syntax format), enter:

   ```
   od -bcx a.out +100.
   ```

   This displays the **a.out** file in octal-byte (**-b**), character (**-c**), and hexadecimal (**-x**) formats, starting from the 100th byte. The **.** (period) after the offset makes it a decimal number. Without the period, the output would start from the 64th (100 octal) byte.

Files

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

Related Information

The **dbx** command, **pg** command.

The National Language Support Overview and Understanding Locale Environment Variables sections in *AIX 5L Version 5.2 National Language Support Guide and Reference*.

---

odmadd Command

Purpose

Adds objects to created object classes.

Syntax

odmadd [*InputFile* ...]

Description

The **odmadd** command takes as input one or more *InputFile* files and adds objects to object classes with data found in the stanza files. Each *InputFile* file is an ASCII file containing the data that describes the objects to be added to object classes. If no file is specified, input is taken from stdin (standard input).

The classes to be added to are specified in the ASCII input file. The file is in the following general format:

```
class1name:
        descriptor1name = descriptor1value
        descriptor2name = descriptor2value
        descriptor3name = descriptor3value
```

```
class2name:
        descriptor4name = descriptor4value
.
.
.
```

The input file can contain the **** (backslash), which is handled as it is in C language. String and method values in the input file must be enclosed in ″ ″ (double-quotation marks). A descriptor value can span more than one line.

Examples

An ASCII input file used by the **odmadd** command looks like the following:

```
Fictional_Characters:
Story_Star       = "Cinderella"
Birthday         = "Once upon a time"
Age              = 19
Friends_of       = Cinderella
Enemies_of       = "Cinderella"

Friend_Table:
Friend_of        = "Cinderella"
Friend           = "Fairy godmother"

Friend_Table:
Friend_of        = "Cinderella"
Friend           = "Mice"

Enemy_Table:
Enemy_of         = "Cinderella"
Enemy            = "Wicked sisters"

Enemy_Table:
Enemy_of         = "Cinderella"
Enemy            = "Mean stepmother"
```

If the preceding file is named `NewObjects`, the following command adds the objects to existing object classes:

```
odmadd NewObjects
```

See ″ODM Example Code and Output″ in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs* for another example of an ASCII file that can be input with the **odmadd** command.

Related Information

Object Data Manager (ODM) Overview for Programmers in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

The **odm_add_obj** subroutine.

List of ODM Commands and Subroutines in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

Understanding ODM Object Classes and Objects in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

How to Create an Object Class in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

How to Add Objects to an Object Class in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

How to Store Object Classes and Objects in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

odmchange Command

Purpose

Changes the contents of a selected object in the specified object class.

Syntax

odmchange -o *ObjectClass* [**-q** *Criteria*] [*InputFile*]

Description

The **odmchange** command, given the object class to modify, the search criteria, and the new object (only for attributes that need to change), modifies all objects that satisfy the search criteria. The *InputFile* file has the same format as the *InputFile* file (the ASCII input file) for the **odmadd** command.

Flags

-o *ObjectClass*	Specifies the object class to modify.
-q *Criteria*	Specifies the criteria used to select objects from the object class. For information on qualifying criteria, see ″Understanding ODM Object Searches″ in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*. If no criteria are specified (no **-q** flag), all object entries in the object class are changed.

Related Information

Object Data Manager (ODM) Overview for Programmers in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

Object Data Manager (ODM) Overview for Programmers in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

The **odmadd** command.

The **odm_change_obj** subroutine.

Understanding ODM Descriptors in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

List of ODM Commands and Subroutines in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

ODM Example Code and Output in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

odmcreate Command

Purpose

Produces the **.c** (source) and **.h** (include) files necessary for ODM application development and creates empty object classes.

Syntax

odmcreate [**-p**] [**-c** | **-h**] *ClassDescriptionFile*

Description

The **odmcreate** command is the ODM class compiler. The command takes as input an ASCII file that describes the objects a user wishes to use in a specific application. The **odmcreate** command can create empty object classes as part of its execution.

The output of the **odmcreate** command is a **.h** file (an include file) that contains the C language definitions for the object classes defined in the ASCII *ClassDescriptionFile* file. The resulting include file is used by the application for accessing objects stored in ODM. The **odmcreate** command also produces a **.c** file that must be compiled and bound in with the application. The **.c** file contains structures and definitions that are used internally by ODM at run time.

The *ClassDescriptionFile* parameter specifies an ASCII file that contains descriptions of one or more object classes. The general syntax for the *ClassDescriptionFile* parameter is as follows:

file : classes
classes : class | classes class
class : head body tail
head : **struct** *ClassName* **{**
tail : **}**
body : elements
elements : elements | elements element
element :**char** *DescriptorName* **[** *DescriptorSize* **];**

vchar *DescriptorName* **[** *DescriptorSize* **];**

binary *DescriptorName* **[** *DescriptorSize* **];**

short *DescriptorName* **;**

long *DescriptorName* **;**

long64 or **int64** or **ODM_LONG_LONG** *DescriptorName* **;**

method *DescriptorName* **;**

link *StdClassName StdClassName ColName DescriptorName* **;**

The default suffix for a *ClassDescriptionFile* file is **.cre**. If no suffix is specified on the **odmcreate** command, then a **.cre** suffix is appended. The file can have C language comments if run with the **-p** flag, and can include **#define** and **#include** lines that can be preprocessed if the **-p** flag is used to run the C language preprocessor on the file.

> **Note:** ODM data bases are 32-bit data bases. The long type, when used in the class description file is a 32-bit data item. The long64 or int64 type, when used in the class description file is a 64-bit data item. The generated files will function the same for both 32- and 64-bit applications.

Flags

-c Creates empty object classes only; does not generate the C language **.h** and **.c** files.
-h Generates the **.c** and **.h** files only; does not create empty classes.
-p Runs the C language preprocessor on the *ClassDescriptionFile* file.

Example

Assuming that a *ClassDescriptionFile* file named `FileName.cre` exists, the following command creates object classes:

```
odmcreate FileName.cre
```

Below is the `FileName.cre` source file and the resulting **.h** file:

```
/* This is an example odmcreate input file */
/* FileName.cre */

       class Class2 {
            char keys[32];
            method card;
            long cash;
            };
      class TstObj {
            long a;
            char b[80];
            link Class2 Class2 card Class2Ln;
            };

/* End of FileName.cre */

/* This is the generated header file FileName.h */
#include <odmi.h>

struct Class2 {
     long _id;         /* unique object id within object class */
     long _reserved;   /* reserved field */
     long _scratch;    /* extra field for application use */
     char keys[32];
     char card[256];   /* method */
     long cash;
     };
#define Class2_Descs 3

extern struct Class Class2_CLASS[];
#define get_Class2_list (a,b,c,d,e) (struct Class2 * ) odm_get_list (a,b,c,d,e)

struct TstObj {
     long _id;         /* unique object id within object class */
     long _reserved;   /* reserved field */
     long _scratch;    /* extra field for application use */
     long a;
     char b[80];
     struct Class2 *Class2Ln;  /* link */
     struct objlistinfo *Class2Ln_info; /* link */
     char Class2Ln_Lvalue[256];        /* link */
     };
#define TstObj_Descs 3

extern struct Class TstObj_CLASS[];
#define get_TstObj_list (a,b,c,d,e) (struct TstObj * ) odm_get_list (a,b,c,d,e)

/* End of generated header file FileName.h */
```

See ″ODM Example Code and Output″ in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs* for another example of a *ClassDescriptionFile* parameter and the resulting **.h** file.

Related Information

Object Data Manager (ODM) Overview in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

The **odm_create_class** subroutine.

List of ODM Commands and Subroutines in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

Understanding ODM Object Classes and Objects in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

Understanding ODM Descriptors in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

How to Create an Object Class in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

How to Add Objects to an Object Class in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

How to Store Object Classes and Objects in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

---

odmdelete Command

Purpose

Deletes selected objects from a specified object class.

Syntax

odmdelete -o *ObjectClass* [**-q** *Criteria*]

Description

The **odmdelete** command, given the object class to delete from and the search criteria, deletes all objects that meet those criteria.

Flags

-o *ObjectClass*	Specifies the object class to delete from.
-q *Criteria*	Specifies the criteria used to select objects from the object class. For information on qualifying criteria, see ″Understanding ODM Object Searches″ in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*. If no criteria are specified (no **-q** flag), then all objects are deleted.

Related Information

Object Data Manager (ODM) Overview for Programmers in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

Object Data Manager (ODM) Overview in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

The **odm_rm_obj** subroutine.

Understanding ODM Object Classes and Objects in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

List of ODM Commands and Subroutines in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

odmdrop Command

Purpose

Removes an object class.

Syntax

odmdrop -o *ClassName*

Description

The **odmdrop** command removes an entire object class and all of its objects. No checking is done to see if other object classes are linked to this one.

Flags

-o *ClassName*	Specifies the object class to remove.

Example

Assuming that an object class named `MyObjectClass` exists, the following command removes the object class:

```
odmdrop -o MyObjectClass
```

Related Information

Object Data Manager (ODM) Overview for Programmers in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

Object Data Manager (ODM) Overview in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

The **odm_rm_class** subroutine.

Understanding ODM Object Classes and Objects in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

List of ODM Commands and Subroutines in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

odmget Command

Purpose

Retrieves objects from the specified object classes into an **odmadd** input file.

Syntax

odmget [**-q** *Criteria*] *ObjectClass* ...

Description

The **odmget** command takes as input a search criteria and a list of object classes, retrieves the selected objects from the specified object classes, and writes an ASCII **odmadd** input file to standard output.

Flags

-q *Criteria* Specifies the search criteria used to select objects from the object class or classes. For information on search criteria, see ″Understanding ODM Object Searches″ in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*. If no criteria are specified (no **-q** flag), all objects in the object class or classes are retrieved.

Example

The following **odmget** command retrieves objects from an existing object class called `Supporting_Cast_Ratings` that has an `Others` descriptor equal to the string `` `Fairy Godmother' ``:

```
odmget -q"Others='Fairy Godmother'" Supporting_Cast_Ratings
```

See the **odmadd** command or ″ODM Example Code and Output″ in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs* for an example of an ASCII **odmadd** input file.

Related Information

The **odmadd** command.

Object Data Manager (ODM) Overview for Programmers in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

Understanding ODM Object Classes and Objects in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

ODM Example Code and Output in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

List of ODM Commands and Subroutines in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

---

odmshow Command

Purpose

Displays an object class definition on the screen.

Syntax

odmshow *ObjectClass*

Description

The **odmshow** command takes as input an object class name (*ObjectClass*) and displays the class description on the screen. The class description is in the format taken as input to the **odmcreate** command.

Example

Assuming that an object class named `MyObjectClass` exists, the following command displays the description of `MyObjectClass` on the screen:

```
odmshow MyObjectClass
```

Also, see the **odmcreate** command or ″ODM Example Code and Output″ in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs* for an example of the output listing.

Related Information

Object Data Manager (ODM) Overview for Programmers in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

The **odmcreate** command.

Understanding ODM Object Classes and Objects in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

ODM Example Code and Output in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

List of ODM Commands and Subroutines in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

---

on Command

Purpose

Executes commands on remote systems.

Syntax

/usr/bin/on [**-i**] [**-d**] [**-n**] *Host Command* [*Argument ...*]

Description

The **on** command executes commands on other systems in an environment that is similar to the one running the program. The **on** command passes the local environment variables to the remote machine, thus preserving the current working directory. When using the **on** command, both users must have the same user identification. Relative path names work only if they are within the current file system. Absolute path names can cause problems since commands are issued at one machine and executed on another.

The standard input is connected to the standard input of the remote command. The standard output and standard error from the remote command are sent to the corresponding files for the **on** command. The root user cannot execute the **on** command.

> **Attention:** When the working directory is remotely mounted over the Network File System (NFS), the Ctrl-Z key sequence causes the window to hang.

Flags

-d Specifies debug mode. Displays status messages as work progresses.

-i Specifies interactive mode. Uses remote echoing and special character processing. This option is needed for programs that expect to be talking to a terminal. All terminal modes and window size changes are increased.

-n Specifies no input. This option causes the remote program to get an end-of-file (EOF) message when it reads from standard input. This flag is necessary when running commands in the background with job control.

Example

To execute the **ls -al** command on another machine and display the in-progress status messages on your terminal, enter:

```
on  -d zorro ls -al
```

In this example, the **on** command executes the **ls** command on a workstation named `zorro`.

Files

/etc/exports
Lists the directories that the server can export.

/etc/inetd.conf
Defines how the **inetd** daemon handles Internet service requests.

Related Information

The **rexd** daemon.

Network File System (NFS) Overview for System Management in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

List of NFS Commands.

---

oslevel Command

Purpose

Reports the latest installed maintenance level of the system.

Syntax

oslevel [**-l** *Level* | **-g** | **-q**] [**-r**] [**-f**]

Description

The **oslevel** command reports the level of the operating system using a subset of all filesets installed on your system. These filesets include the Base Operating System (BOS), base devices, base printers, and X11 2d.

The **oslevel** command also prints information about maintenance levels, including which filesets are not at a specified maintenance level.

Flags

-l *Level*	Lists file sets at levels earlier than maintenance level specified by the *Level* parameter.
-f	Forces the **oslevel** command to rebuild the cache for this operation.
-g	Lists file sets at levels later than the current maintenance level.
-q	Lists names of known maintenance levels that can be specified using the **-l** flag.
-r	Applies all flags to the recommended maintenance levels.

Examples

1. To determine the maintenance level of the system, type:

   ```
   oslevel
   ```

2. To determine the file sets that are below level 4.1.2.0, type:

   ```
   oslevel -l 4.1.2.0
   ```

3. To determine the file sets at levels later than the current maintenance level, type:

   ```
   oslevel -g
   ```

4. To determine the highest recommended maintenance level reached for the current version of AIX on the system, type:

```
oslevel -r
```

5. To list all known recommended maintenance levels on the system, type:

```
oslevel -rq
```

Files

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

Related Information

The **lslpp** command.

---

ospf_monitor Command

Purpose

Monitors the OSPF gateways.

Syntax

ospf_monitor *mon_db_file*

Description

The **ospf_monitor** command is used to query OSPF routers. The **ospf_monitor** command operates in interactive mode. It allows the user to query the various OSPF routers to provide detailed information on I/O statistics, error logs, link-state data bases, AS external data bases, the OSPF routing table, configured OSPF interfaces, and OSPF neighbors.

Specify the complete pathname of a database composed of records configuring destinations for **ospf_monitor** remote commands with *mon_db_file*. Each destination record is a single-line entry which lists the destination IP address, the destination hostname, and an OSPF authentication key (if authentication is activated by the destination). Since authentication keys may be present in the destination records, it is recommended that general access to this database be restricted.

Refer to RFC-1583 (OSPF Specification, version 2) for details about OSPF database and packet formats.

Commands

Upon entering interactive mode, **ospf_monitor** presents the `'[ # ] dest command params >'` prompt, at which you can enter any of **ospf_monitor**'s interactive commands. Interactive commands can be interrupted at any time with a keyboard interrupt.

Note: The command line length must be less than 200 characters.

Local Commands

?	Displays all local commands and their functions.
?R	Displays all remote commands and their functions.
d	Displays all configured destinations. This command displays *dest_index* , the IP address, and the hostname of all potential **ospf_monitor** command destinations configured in *mon_db_file*.
h	Displays the command history buffer showing the last 30 interactive commands.
x	Exits the **ospf_monitor** program.
@ *remote_command*	Sends *remote_command* to the same (previous) destination.
@*dest_index remote_command*	Sends *remote_command* to configured destination *dest_index*.

F *filename* Sends all **ospf_monitor** output to *filename*.

S Sends all **ospf_monitor** output to stdout.

Remote Commands

a *area_id type ls_id adv_rtr* Displays link state advertisement. *Area_id* is the OSPF area for which the query is directed. *adv_rtr* is the router-id of the router which originated this link state advertisement. *Type* specifies the type of advertisement to request and should be specified as follows:

- **1** Request the router links advertisements. They describe the collected states of the router's interfaces. For this type of request, the *ls_id* field should be set to the originating router's Router ID.
- **2** Request the network links advertisements. They describe the set of routers attached to the network. For this type of request, the *ls_id* field should be set to the IP interface address of the network's Designated Router.
- **3** Request the summary link advertisements describing routes to networks. They describe inter-area routes, and enable the condensing of routing information at area borders. For this type of request, the *ls_id* field should be set to the destination network's IP address.
- **4** Request the summary link advertisements describing routes to AS boundary routers. They describe inter-area routes, and enable the condensing of routing information at area borders. For this type of request, the *ls_id* field should be set to the Router ID of the described AS boundary router.
- **5** Request the AS external link advertisements. They describe routes to destinations external to the Autonomous System. For this type of request, the *ls_id* field should be set to the destination network's IP address.

c Displays cumulative log. This log includes input/output statistics for monitor request, hello, data base description, link-state request, link-state update, and link-state ack packets. Area statistics are provided which describe the total number of routing neighbors and number of active OSPF interfaces. Routing table statistics are summarized and reported as the number of intra-area routes, inter-area routes, and AS external data base entries.

e Displays cumulative errors. This log reports the various error conditions which can occur between OSPF routing neighbors and shows the number of occurrences for each.

h Displays the next hop list. This is a list of valid next hops mostly derived from the SPF calculation.

l [*retrans*] Displays the link-state database (except for ASE's). This table describes the routers and networks making up the AS. If *retrans* is non-zero, the retransmit list of neighbors held by this lsdb structure will be printed.

A [*retrans*] Displays the AS external data base entries. This table reports the advertising router, forwarding address, age, length, sequence number, type, and metric for each AS external route. If *retrans* is non-zero, the retransmit list of neighbors held by this lsdb structure will be printed.

o [*which*] Displays the OSPF routing table. This table reports the AS border routes, area border routes, summary AS border routes, networks, summary networks and AS external networks currently managed via OSPF. If *which* is omitted, all of the above will be listed. If specified, the value of *which* (between 1 and 63) specifies that only certain tables should be displayed. The appropriate value is determined by adding up the values for the desired tables from the following list:

1 Routes to AS border routers in this area.

2 Routes to area border routers for this area.

4 Summary routes to AS border routers in other areas.

8 Routes to networks in this area.

16 Summary routes to networks in other areas.

32 AS routes to non-OSPF networks.

I Displays all interfaces. This report shows all interfaces configured for OSPF. Information reported includes the area, interface IP address, interface type, interface state, cost, priority, and the IP address of the DR and BDR for the network.

N Displays all OSPF routing neighbors. Information reported includes the area, local interface address, router ID, neighbor IP address, state, and mode.

V Displays Gated version information.

Related Information

The **gated** Daemon.

---

pac Command

Purpose

Prepares printer/plotter accounting records.

Syntax

/usr/sbin/pac [**-c**] [**-m**] [**-p***Price*] [**-P***Printer*] [**-q***File*] [**-r**] [**-s**] [*Name ...*]

Description

The **pac** command prepares printer/plotter accounting records for each user of the selected printer or for the users specified by the *Name* parameter. For printer choices, see the **-P** flag.

The unit of measure is the number of pages, with the exception of raster devices, for which feet of paper is measured. Output is expressed both as the number of units used and the charge in dollars. For information on the charge (price) per unit, see the **-p** flag.

The accounting file specified in the **/etc/qconfig** file and the file created to contain the summary information must grant read and write permissions to the root user or printq group. The **pac** command generates the summary file name by appending **_sum** to the path name specified by the `acctfile =` clause in the **/etc/qconfig** file. For example, if the **qconfig** file reads:

```
acctfile = /var/adm/1p0acct
```

The **pac** command expects the summary file to be named **/var/adm/1p0acct_sum**.

Flags

-c Sorts the output by price instead of alphabetically by user.

-m Groups all the printing charges for a user, regardless of the host machine.

-p*Price* Specifies the price, in dollars, charged per unit of output. By default, the system charges $0.02 per unit.

-P*Printer* Specifies the printer for which accounting records are prepared. By default, the system selects the printer named by the **PRINTER** environment variable or the default value **lp0**.

> **Note:** When the **LPDEST** environment variable is set, it takes precedence over the **PRINTER** environment variable, which has an identical function. Any destination options issued from the command line override both the **LPDEST** and **PRINTER** environment variables.

-q*File* Specifies the queue configuration file. The default value is the **/etc/qconfig** file.

-r Reverses the sorting order, so that records are sorted alphabetically from z to a, or in descending order by price.

-s Summarizes the accounting information in a summary file. This flag is needed for busy systems.

Examples

1. To produce printer/plotter accounting information for all users of the `lp0` printer, enter:

   ```
   /usr/sbin/pac
   ```

   The command displays the number of printed pages and the charge, sorted by user. This example assumes that there is no **PRINTER** environment variable.
2. To collect printer/plotter accounting records in a summary file, enter:

   ```
   /usr/sbin/pac  -s
   ```
3. To produce printer/plotter accounting information for `smith`, `jones`, and `greene` from the `lp12` printer, enter:

   ```
   /usr/sbin/pac  -Plp12 smith jones greene
   ```

   > **Note:** Do not place a space between a flag and its variable; for example, the **-p***Price,* **-P***Printer*, and **-q***File.*

Files

/usr/sbin/pac	Contains the **pac** command.
/etc/qconfig	Specifies the path to the file.

Related Information

The **acctcms** command, **acctcom** command, **acctcon1** or **acctcon 2** command, **acctmerg** command, **acctprc1**, **acctprc2**, or **accton** command, **runacct** command.

The **qconfig** file.

Setting Up an Accounting System in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*.

Accounting Overview in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*.

---

pack Command

Purpose

Compresses files.

Syntax

pack [**-f**] [**-**] *File ...*

Description

The **pack** command stores the file specified by the *File* parameter in a compressed form. The input file is replaced by a packed file with the same name and the suffix **.z** appended. The packed file maintains the same access modes, access and modification dates, and owner as the original file. The input file name can contain no more than 253 bytes to allow space for the added **.z** suffix. If the **pack** command is successful, the original file is removed. Packed files can be restored to their original form using the **unpack** or **pcat** commands.

If the **pack** command cannot create a smaller file, it stops processing and reports that it is unable to save space. (A failure to save space generally happens with small files or files with uniform character distribution.) The amount of space saved depends on the size of the input file and the character frequency distribution. Because a decoding tree forms the first part of each **.z** file, you do not save space with files smaller than three blocks. Typically, text files are reduced 25 to 40 percent.

> **Note:** The **pack** command's algorithm uses Huffman encoding to compress files. This algorithm has fundamental limitations. Consequently, the **pack** command can only consistently compress files under 8MB. To consistently compress files larger than 8MB, use the **compress** command.

The exit value of the **pack** command is the number of files that it could not pack. The **pack** command does not pack under any of the following conditions:

- The file is already packed.
- The input file name has more than 253 bytes.
- The file has links.
- The file is a directory.
- The file cannot be opened.
- No storage blocks are saved by packing.
- A file called *File***.z** already exists.
- The **.z** file cannot be created.
- An I/O error occurred during processing.

Flags

-f Forces packing of the file specified by the *File* parameter. This is useful for packing an entire directory, even if some of the files will not benefit.

Parameters

File Specifies the file to be packed.

- Displays statistics about the file specified by the *File* parameter. The statistics are calculated from a Huffman minimum redundancy code tree built on a byte-by-byte basis. Additional occurrences of the **-** (minus sign) parameter on the command line toggles this function for the next specified file. See example 2.

Exit Status

This command returns the following exit values:

0 Specifies that the file was successfully packed.

>0 Specifies that an error occurred.

Examples

1. To compress the files named `chap1` and `chap2` and display the revised file names, enter:

   ```
   pack chap1 chap2
   ```

   The compressed versions are renamed `chap1.z` and `chap2.z`. The **pack** command displays the percent decrease in size for each file compressed.
2. To display statistics about the amount of compression done, enter:

   ```
   pack  - chap1  - chap2
   ```

   This compresses the files named `chap1` and `chap2` and displays statistics about the file named `chap1`, but not about the file named `chap2`. The first **-** (minus sign) parameter turns on the statistic display, and the second **-** parameter turns it off.

Files

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

Related Information

The **cat** command, **pcat** command, **unpack** command.

Files Overview in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices*.

Input and Output Redirection Overview in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices*.

---

packf Command

Purpose

Compresses the contents of a folder into a file.

Syntax

packf [**+***Folder*] [*Messages*] [**-file** *File*]

Description

The **packf** command compresses the messages in a folder into a specified file. By default, the **packf** command compresses messages from the current folder and places them in the **msgbox** file. If the file does not exist, the system prompts you for permission to create it. Each message in the file is separated with four Ctrl-A characters and a new-line character.

Note: You can use the **inc** command to unpack compressed messages.

Flags

-file *File*	Specifies the file in which to put compressed messages. The default is the **./msgbox** file. If the file exists, the **packf** command appends the messages to the end of the file. Otherwise, the system prompts you for permission to create the file.
+*Folder*	Identifies the folder containing the messages you want to pack.
-help	Lists the command syntax, available switches (toggles), and version information.

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

Messages Specifies what messages to pack. The *Messages* parameter can specify several messages, a range of messages, or a single message. If several messages are specified, the first message packed becomes the current message. Use the following references to specify messages:

Number
Number of the message. When specifying several messages, separate each number with a comma. When specifying a range, separate the first and last numbers in the range with a hyphen.

Sequence
A group of messages specified by the user. Recognized values include:

all All the messages in the folder. This is the default.

cur or . (period)
Current message.

first First message in a folder.

last Last message in a folder.

next Message immediately after the current message.

prev Message immediately before the current message.

Profile Entries

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

`Current-Folder:`	Sets your default current folder.
`Msg-Protect:`	Sets the protection level for your new message files.
`Path:`	Specifies the user's MH directory.

Examples

1. To pack all the messages in the current folder and place the resulting text in the **schedule** file, enter:

   ```
   packf  -file schedule
   ```

   The system responds with a message similar to the following:

   ```
   Create file "/home/mary/schedule"?
   ```

   Enter `y` to create the file.
2. To pack the range of messages from `3` to `7` from the **test** folder into an existing **msgbox** file, enter:

   ```
   packf  +test 3-7
   ```

   The system responds with the `shell` prompt when the command is complete.
3. To pack the current, first, and last message in the **inbox** folder into an existing **msgbox** file, enter:

   ```
   packf cur first last
   ```

Files

$HOME/.mh_profile	Specifies the MH user profile.
/usr/bin/packf	Contains the **packf** command.

Related Information

The **inc** command.

The **.mh_alias** file format, **.mh_profile** file format.

Mail Overview in *AIX 5L Version 5.2 System User's Guide: Communications and Networks*.

---

pagdel Command

Purpose

Removes any existing PAG association within the current process' credentials.

Syntax

paginit [**-R** *module_name*] [*username*]

Description

The **pagdel** command will remove the PAG identifier from the current process' credentials structure. If the **-R** option is omitted, the registry attribute will be used as the **module_name**.

Flags

-R *module_name* Specifies a load module found in **/usr/lib/security/modules.cfg**. The **load_module** will be asked to delete any PAG currently associated with the process.

Security

Access Control: This command should grant execute (x) access only to the `root` user and members of the security group. This command should be installed as a program in the trusted computing base (TCB). The command should be owned by the `root` user with the **setuid** (SUID) bit set.

Auditing

USER_PagDelete

Example

To remove the PKI authentication certificate associated with the current process, type:

```
pagdel -R FPKI
```

Related Information

The **paginit** and **paglist** commands.

---

pagesize Command

Purpose

Displays the system page size.

Syntax

pagesize

Description

The **pagesize** command prints the size, in bytes, of a page of memory, as returned by the **getpagesize** subroutine. Provided for system compatibility, this command is useful when constructing portable shell scripts.

Example

To obtain the size system page, enter:

```
pagesize
```

The system returns the number of bytes, such as 4096.

Files

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

Related Information

The **getpagesize** subroutine.

---

paginit Command

Purpose

Authenticate a user and create a PAG association.

Syntax

paginit [**-R** *module_name*] [*username*]

Description

The **paginit** command authenticates *username* (by default, the user issuing the command) and creates an association between the *username* and a kernel token called a Process Authentication Group entry (PAG). A new login shell is spawned by this command.

If the **-R** flag is not given, **paglist** queries the user's registry attribute and use that value for *module_name*.

To associate the *username* with an alternate Identification and Authentication (I&A) mechanism, the **-R** flag can be used to specify the I&A load module used to create the user. Load modules are defined in the **/usr/lib/security/methods.cfg** file.

Flags

-R *module_name*	Specifies the loadable I&A module used to authenticate the user.

Parameters

username	Specifies the user. This parameter defaults to the user issuing the command. Only the **root** user may override the default.

Security

Access Control: This command should be executable by all. It should be owned by **root** and should be **setuid**.

Auditing

USER_Paginit

Example

```
paginit -R FPKI
```

The user is authenticated using the registry FPKI, which is defined in the **/usr/lib/security/methods.cfg** file. A PAG is associated with the current process credentials.

Related Information

The **pagdel** and **paglist** commands.

---

paglist Command

Purpose

Lists authentication information associated with the current process.

Syntax

paglist [**-R** *module_name*]

Description

The **paglist** command queries the current process' credentials to display its authentication certificate.

If the **-R** option is not given, **paglist** will query the user's registry attribute and use that value for **module_name**.

Flags

-R *module_name*	Specifies that the load module *module_name* is to list its authentication certificate associated with the current process.

Security

Access Control: This command runs with the ID of the invoking user, without any elevated privileges. It should be owned by `root`, but executable by all.

Example

```
paglist -R FPKI
```

This example will list the PAG associated with the current process within the FPKI registry.

Related Information

The **pagdel** and **paginit** commands.

---

panel20 Command

Purpose

Diagnoses activity between an HIA and the 5080 Control Unit.

Syntax

panel20 [**HIA0** | **HIA1** | **HIA2**]

Description

Use the **panel20** command as a diagnostic tool to determine whether the Host Interface Adapter (HIA) is correctly installed and communicating with the 5088 Graphics Channel Control Unit (GCCU).

The **panel20** command displays a diagnostic screen with the following columns: `Device Name`, `Channel Address`, `Link Address`, `Link Status`, `Poll Counter`, `SNRM Counter`.

If the HIA is correctly installed and the host operating system is correctly configured to support 3270 devices on the 5088, the entries in the Set Normal Response Mode (`SNRM Counter`) column will be increasing. If the entries in `SNRM Counter` are not increasing, refer to problem determination procedures for the HIA and verify that the host operating system is correctly configured.

Examples

To start the **panel20** command, enter:

```
panel20
```

By default, the **panel20** command will monitor HIA0. To monitor HIA1 or HIA2, enter:

```
panel20 HIA1
```

OR

```
panel20 HIA2
```

---

passwd Command

Purpose

Changes a user's password.

Syntax

passwd [**-R** *load_module*] [**-f** | **-s**] [*User*]

Description

The **passwd** command sets and changes passwords for users. Use this command to change your own password or another user's password. You can also use the **passwd** command to change the full name (gecos) associated with your login name and the shell you use as an interface to the operating system.

Depending on how the user is defined, the user's password can exist locally or remotely. Local passwords exist in the **/etc/security/passwd** database. Remote passwords exist in the Network Information Service (NIS) or Distributed Computing Environment (DCE) database.

To change your own password, enter the **passwd** command. The **passwd** command prompts the nonroot user for the old password (if one exists) and then prompts for the new password twice. (The password is never displayed on the screen.) If the two entries of the new password do not match, the **passwd** command prompts for the new password again.

Note: The **passwd** command uses only the first eight characters of your password for local and NIS passwords. Only 7-bit characters are supported in passwords. For this reason, National Language Support (NLS) code points are not allowed in passwords.

To change another user's password, enter the **passwd** command and the user's login name (the *User* parameter). Only the root user or a member of the security group is permitted to change the password for another user. The **passwd** command prompts you for the old password of the user as well as the new password. However, the **passwd** command does not prompt the root user for either the old user password or the root password or enforce any password restrictions upon the root user.

The **/etc/passwd** file records your full name and the path name of the shell that you use. To change your recorded name, enter the **passwd -f** command. To change your login shell, enter the **passwd -s** command.

Construct locally-defined passwords according to the password restrictions in the **/etc/security/user** configuration file. This file contains the following restrictions:

minalpha	Specifies the minimum number of alphabetic characters.
minother	Specifies the minimum number of other characters.
minlen	Specifies the minimum number of characters. **Note:** This value is determined by either the `minalpha` value plus the `minother` value or the `minlen` value, whichever is greater.
mindiff	Specifies the minimum number of characters in the new password that are not in the old password. **Note:** This restriction does not consider position. If the new password is `abcd` and the old password is `edcb`, the number of different characters is 1.
maxrepeats	Specifies the maximum number of times a single character can be used in a password.
minage	Specifies the minimum age at which a password can be changed. Passwords must be kept for a minimum period. This value is measured in weeks.
maxage	Specifies the maximum age of a password. A password must be changed after a specified amount of time measured in weeks.
maxexpired	Specifies the maximum number of weeks beyond the `maxage` value that a password can be changed by the user.
histexpire	Specifies the number of weeks that a user cannot reuse a password.
histsize	Specifies the number of previous passwords that the user cannot reuse.
dictionlist	Specifies the list of dictionary files checked when a password is changed.
pwdchecks	Specifies the list of external password restriction methods invoked when a password is changed.

If the root user adds the **NOCHECK** attribute to your flags entry in the **/etc/security/passwd** file, your password does not need to meet these restrictions. Also, the root user can assign new passwords to other users without following the password restrictions.

If the root user adds the **ADMIN** attribute to your flags entry or if the **password** field in the **/etc/passwd** file contains an * (asterisk), only the root user can change your password. The root user also has the exclusive privilege of changing your password if the **password** field in **/etc/passwd** contains an ! (exclamation point) and the **password** field in the **/etc/security/passwd** file contains an * (asterisk).

If the root user changes your password, the **ADMCHG** attribute is automatically added to your flags entry in the **/etc/security/passwd** file. In this case, you must change the password the next time you log in.

If the user's **registry** value in the **/etc/security/user** file is either DCE or NIS, the password change can only occur in the specified database.

Flags

-f Changes the user information accessed by the **finger** command. You can use this flag to provide your full name in the **/etc/passwd** file.

-s Changes the login shell.

-R Specifies the loadable I&A module used to change a user's password.
load_module

Security

Access Control: Install this program as part of the Trusted Computing Base with execute access for all users. In addition, this program should run **setuid** to the root user to get write access to the password files.

Files Accessed:

Mode	File
r	**/etc/security/user**
rx	**/usr/lib/security/DCE**
rx	**/usr/lib/security/NIS**
rw	**/etc/passwd**
rw	**/etc/security/passwd**
r	**/etc/security/login.cfg**

Auditing Events:

Event	Information
PASSWORD_Change	user

Examples

1. To change your password, type:

   ```
   passwd
   ```

   The **passwd** command prompts you for your old password, if it exists and you are not the root user. After you enter the old password, the command prompts you twice for the new password.
2. To change your full name in the **/etc/passwd** file, type:

   ```
   passwd  -f
   ```

   The **passwd** command displays the name stored for your user ID. For example, for login name `sam`, the **passwd** command could display this message:

   ```
   sam's current gecos:
               "Sam Smith"
   Change (yes) or no)? >
   ```

   If you type a `Y` for yes, the **passwd** command prompts you for the new name. The **passwd** command records the name you enter in the **/etc/passwd** file.
3. To use a different shell the next time you log in, type:

   ```
   passwd - s
   ```

   The **passwd** command lists the path names of the available shells and the shell you are currently using. The command also displays a prompt:

   ```
   Change (yes) or (no)? >
   ```

   If you type a `Y` for yes, the **passwd** command prompts you for the shell to use. The next time you log in, the system provides the shell that you specify here.

Files

/usr/bin/passwd	Contains the **passwd** command.
/etc/passwd	Contains user IDs, user names, home directories, login shell, and finger information.
/etc/security/passwd	Contains encrypted passwords and security information.

Related Information

The **chfn** command, **chsh** command, **login** command, **pwdadm** command, **pwdck** command.

The **getpass** subroutine, **newpass** subroutine.

Users, Groups, and Passwords in *AIX 5L Version 5.2 Security Guide*.

Installing and Configuring the Trusted Computing Base in *AIX 5L Version 5.2 Security Guide*.

Shells Overview in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices* to learn more about the available shells.

---

paste Command

Purpose

Joins lines from one or more files.

Syntax

paste [**-s**] [**-d** *List*] *File1* ...

Description

The **paste** command reads input from the files specified on the command line. The command reads from standard input if a **-** (minus sign) appears as a file name. The command concatenates the corresponding lines of the given input files and writes the resulting lines to standard output.

By default, the **paste** command treats each file as a column and joins them horizontally with a tab character (parallel merging). You can think of the **paste** command as the counterpart of the **cat** command (which concatenates files vertically, that is, one file after another).

With the **-s** flag, the **paste** command combines subsequent lines of the same input file (serial merging). These lines are joined with the tab character by default.

Notes:

1. The **paste** command supports up to 32767 input files (the **OPEN_MAX** constant).
2. The action of the **pr -t -m** command is similar to that of the **paste** command, but creates extra spaces, tabs, and lines for a nice page layout.
3. Input files should be text files, but may contain an unlimited number of line lengths.

Flags

-d *List* Changes the delimiter that separates corresponding lines in the output with one or more characters specified in the *List* parameter (the default is a tab). If more than one character is in the *List* parameter, then they are repeated in order until the end of the output. In parallel merging, the lines from the last file always end with a new-line character instead of one from the *List* parameter.

The following special characters can also be used in the *List* parameter:

\n New-line character

\t Tab

**** Backslash

\0 Empty string (not a null character)

c An extended character

You must put quotation marks around characters that have special meaning to the shell.

-s Merges subsequent lines from the first file horizontally. With this flag, the **paste** command works through one entire file before starting on the next. When it finishes merging the lines in one file, it forces a new line and then merges the lines in the next input file, continuing in the same way through the remaining input files, one at a time. A tab separates the lines unless you use the **-d** flag. Regardless of the *List* parameter, the last character of the file is forced to be a new-line character.

Exit Status

This command returns the following exit values:

0 Successful completion.

>0 An error occurred.

Examples

1. To paste several columns of data together, enter:

   ```
   paste names places dates > npd
   ```

   This creates a file named `npd` that contains the data from the `names` file in one column, the `places` file in another, and the `dates` file in a third. If the `names`, `places`, and `dates` file look like:

   ```
   names           places          dates
   rachel          New York        February 5
   jerry           Austin          March 13
   mark            Chicago         June 21
   marsha          Boca Raton      July 16
   scott           Seattle         November 4
   ```

   then the `npd` file contains:

   ```
   rachel          New York        February 5
   jerry           Austin          March 13
   mark            Chicago         June 21
   marsha          Boca Raton      July 16
   scott           Seattle         November 4
   ```

   A tab character separates the name, place, and date on each line. These columns do not always line up because the tab stops are set at every eighth column.

2. To separate the columns with a character other than a tab, enter:

   ```
   paste  -d"!@" names places dates > npd
   ```

This alternates ! and @ as the column separators. If the `names`, `places`, and `dates` files are the same as in example 1, then the `npd` file contains:

```
rachel!New York@February 5
jerry!Austin@March 13
mark!Chicago@June 21
marsha!Boca Raton@July 16
scott!Seattle@November 4
```

3. To display the standard input in multiple columns, enter:

```
ls | paste - - - -
```

This lists the current directory in four columns. Each - (minus) tells the **paste** command to create a column containing data read from the standard input. The first line is put in the first column, the second line in the second column, and so on.

This is equivalent to:

```
ls | paste  -d"\t\t\t\n"  -s s
```

This example fills the columns across the page with subsequent lines from the standard input. The `-d"\t\t\t\n"` defines the character to insert after each column: a tab character (`\t`) after the first three columns, and a new-line character (`\n`) after the fourth. Without the **-d** flag, the **paste -s -** command would display all of the input as one line with a tab character between each column.

Files

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

Related Information

The **cat** command, **cut** command, **grep** command, **pr** command.

National Language Support Overview in *AIX 5L Version 5.2 National Language Support Guide and Reference*.

Files Overview in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices*.

Input and Output Redirection Overview in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices*.

---

patch Command

Purpose

Applies changes to files.

Syntax

patch [**-b** [**-B** *Prefix*]] [**-f**] [**-l**] [**-N**] [**-R**] [**-s**] [**-v**] [**-c** | **-e** | **-n**] [**-d** *Directory*] [**-D** *Define*] [**-F** *Number*] [**-i** *PatchFile*] [**-o** *OutFile*] [**-p** *Number*] [**-r** *RejectFile*] [**-x** *Number*] [*File*]

Description

The **patch** command reads a source file's instructions on how to change a file, then applies the changes. The source file contains difference listings (or *diff* listings) produced by the **diff** command. The diff listings

are the result of comparing two files and building instructions about how to reconcile the differences. By default, the **patch** command uses the source file read from standard input, but this can be overridden using the **-i** flag and the *PatchFile* variable.

Diff listings come in three formats: normal, context, or in the style of the ed editor. The **patch** command determines the diff listing format, unless overruled by the **-c**, **-e**, or **-n** flags.

By default, the patched version of a file replaces the original version. When the **-b** flag is specified, the original of each patched file is saved in a file of the same name with the suffix **.orig** appended to it. You may also specify where you want the output to go with the **-o** flag.

The **-p** flag makes it possible to customize a patch file to local user directory structures without manually editing the patch file. This is done by specifying how many components to remove from the full path name. For example, if the patch file contained the path name `/curds/whey/src/blurfl/blurfl.c`, then:

- **-p 0** causes the entire path name to be used.
- **-p 1** removes the leading slash, leaving `curds/whey/src/blurfl/blurfl.c`.
- **-p 4** removes leading slashes and the first three directories, leaving `blurfl/blurfl.c`.

Not specifying the **-p** flag causes the **patch** command to use the base name. In the above example, that would be `blurfl.c`.

Patch File Format

The patch file must contain one or more lines of header information followed by one or more patches. Each patch must contain one or more lines of file name identification in the format produced by the **diff -c** command, and one or more sets of **diff** command output, customarily called *hunks*.

The **patch** command skips any leading text in a patch file, applies the actual diff listing, and skips any trailing text. Thus, you could use as a patch file an article or message that includes a diff listing, and the **patch** command would still work. In such a case, if the entire diff listing is indented by a consistent amount, the **patch** command will also adjust for that spacing.

To change a line range within the original file, each hunk within a patch must be a separate diff listing. The line numbers for successive hunks within a patch must occur in ascending order.

File Name Determination

If no *File* parameter is specified, the **patch** command performs the following steps to obtain the name of the file to edit:

1. In the header of a context diff listing, the file name is determined from lines beginning with *** (three asterisks) or —- (three dashes). A line beginning with *** indicates the name of the file from which the patches were taken, while a line beginning with —- indicates the name of the file to which the patches should be applied. The shortest name of an existing file is selected.
2. If there is an `Index:` line in the leading text, the **patch** command tries to use the file name from that line.
3. A context diff header takes precedence over an `Index:` line.
4. If no file name can be determined from the leading text, the **patch** command prompts you for the name of the file to patch.
5. If the original file cannot be found, but a suitable SCCS or RCS file is available, the **patch** command attempts to get or check out the file.
6. If the leading text contains a `Prereq:` line, the **patch** command takes the first word from the prerequisites line (normally a version number) and checks the input file to see if that word can be found. If not, the **patch** command prompts you for confirmation before proceeding.

Patch Application

If the patch file contains more than one patch, the **patch** command tries to apply each diff listing as if it came from a separate patch file. In this case, the name of the file to patch is determined for each diff listing, and the header text before each diff listing is examined for information such as file name and revision level.

If you specify the **-c**, **-e**, or **-n** flag, the **patch** command interprets information within each hunk as a context difference, an ed editor difference, or a normal difference, respectively. Otherwise, the **patch** command determines the type of difference based on the format of the information within the hunk.

The **patch** command searches for the place to apply each hunk by taking the first line number of the hunk and adding or subtracting any line offset caused by applying the previous hunk. If an exact match is not possible at this line location, the **patch** command scans both forward and backward for a set of lines matching the hunk's content exactly.

If no such place is found, and if the **patch** command is applying a context diff listing, the **patch** command can search for a less exact match. A *fuzz factor* specifies how many lines can be inexactly matched. If the fuzz factor is set to 1 or more, the **patch** command performs a second scan, this time ignoring the first and last line of context. If no match results, and the maximum fuzz factor is set to 2 or more, the **patch** command performs a third scan, this time ignoring the first two lines and the last two lines of the context. (The default maximum fuzz factor is 2.) If no match is found, the **patch** command places the hunk in a reject file. The reject file is created with the same name as the output file and the suffix **.rej**. This naming convention can be overridden by using the **-r** flag.

The rejected hunk is written in context diff listing form, regardless of the format of the patch file. If the input was a normal or ed editor style difference, the reject file may contain differences with zero lines of context format. The line numbers on the hunks in the reject file may be different from the line numbers in the patch file. This is because the reject file line numbers reflect the approximate locations for the failed hunks in the new file rather than the old one.

As each hunk is completed, the **patch** command tells you whether the hunk succeeded or failed. You are also informed of the new line number assumed for each hunk. If this is different from the line number specified in the diff listing, you are notified of the offset. The **patch** command also tells you if a fuzz factor was used to make the match.

> **Note:** A single large offset may be an indication that a hunk was installed in the wrong place. Use of a fuzz factor may also indicate bad placement.

Preparing Patches for Other Users

Programmers preparing patches that will be shipped to other users should consider the following additional guidelines:

- If you try to apply the same patch twice, the **patch** command assumes the second application should be a reverse patch and prompts you for confirmation of this reversal. Therefore, avoid sending out reversed patches, since this makes users wonder whether they already applied the patch.
- It is recommended that you keep a **patchlevel.h** file that is updated with the latest patch level. The patch level can then be used as the first diff listing in the patch file you send out. If your patch includes a `Prereq:` line, users cannot apply patches out of order without receiving a warning.
- Make sure you specify the file names correctly, either in a context diff listing header or with an `Index:` line. If you are patching something in a subdirectory, be sure to tell the patch user to specify a **-p** flag as needed.
- You can create a file by sending out a diff listing that compares a null file to the file you want to create. However, this only works if the file you want to create does not already exist in the target directory.
- While you may be able to put many diff listings into one file, it is advisable to group related patches into separate files.

- The **patch** command cannot tell if the line numbers are incorrect in an ed script, and can only detect bad line numbers in a normal diff listing when it finds a change or a delete command. A context diff listing using a fuzz factor of 3 may have the same line-number problem. Until a suitable interactive interface is added, use a context diff listing in such cases to check the changes for accuracy. Compilation without errors usually means that the patch worked, but it is not an infallible indicator.
- The results of the **patch** command are guaranteed only when the patch is applied to exactly the same version of the file from which the patch was generated.
- If the code has been duplicated, for example:

```
#ifdef
... NEWCODE
#else
... OLDCODE
# endif
```

  the **patch** command is incapable of patching both versions. If the **patch** command succeeds, it may have patched the wrong version and return a successful exit status.

Flags

-b Saves a copy of each modified file before the differences are applied. The copied original is filed with the same name and the suffix **.orig**. If a file by that name already exists, it is overwritten. If multiple patches are applied to the same file, only one copy is made of the original file at the time of the first patch. If the **-o** *OutFile* flag is also specified, the **.orig** file is not created. But if the specified out file already exists, *OutFile***.orig** is created.

-B *Prefix* Specifies a prefix to the backup file name. This flag only works in conjunction with the **-b** flag.

-c Interprets the patch file as a context diff listing (the output of the **diff -c** or **diff -C** command). This flag cannot be used with the **-e** or **-n** flag.

-d *Directory* Changes the current directory to the specified directory before processing.

-D *Define* Marks changes with the following C preprocessor construct:

```
#ifdef Define
...  (NEWCODE)
#else
...  (OLDCODE)
#endif /* Define */
```

The *Define* variable is used as the differentiating symbol. This flag only works when the normal or context form of diff listing is used as a patch file.

-e Interprets the patch file as an ed editor script. This flag cannot be used with the **-c** or **-n** flag.

-f Suppresses queries to the user. To suppress commentary, use the **-s** flag.

-F *Number* Sets the maximum fuzz factor. This flag applies to context diff listings only and causes the **patch** command to ignore the specified number of lines when determining where to install a hunk. If the **-F** flag is not specified, the default fuzz factor is 2. The factor may not be set to more than the number of lines of content in the context diff listing (ordinarily 3).

Note: A larger fuzz factor increases the odds of a faulty patch.

-i *PatchFile* Reads the patch information from the specified file, rather than from standard input.

-l (lowercase L) Causes any sequence of blank characters in the diff listing script to match any sequence of blank characters in the input file. Other characters are matched exactly.

-n Interprets the script as a normal diff listing. This flag cannot be used with the **-c** or **-e** flag.

-N Ignores patches where the differences have already been applied to the file. By default, already-applied patches are rejected.

-o *OutFile* Copies the files to be patched, applies the changes, then writes the modified version to the specified output file. Multiple patches for a single file are applied to the intermediate versions of the file created by any previous patches. Therefore, multiple patches result in multiple, concatenated versions of the output file.

-p *Number* Sets the path name strip count, which controls how path names found in the patch file are treated. This flag is useful if you keep your files in a directory different from the specified path. The strip count specifies how many slashes are stripped from the front of the path name. Any intervening directory names are also stripped. For example, assume a patch file specified `/u/leon/src/blurf1/blurf1.c`:

- **-p 0** leaves the entire path name unmodified.
- **-p 1** removes the leading slash, leaving `u/leon/src/blurf1/blurf1.c`.
- **-p 4** removes four slashes and three directories, leaving `blurf1/blurf1.c`.

If the **-p** flag is not specified, only the base name (the final path name component) is used. This flag works only when the *File* parameter is not specified.

-r *RejectFile* Overrides the default reject file name. The default reject file name is formed by appending the suffix **.rej** to the original file name.

-R Reverses the sense of the patch script. For example, if the diff listing was created from new version to old version, using the **-R** flag causes the **patch** command to reverse each portion of the script before applying it. Rejected differences are saved in swapped format. The **-R** flag cannot be used with ed scripts, because there is too little information to reconstruct the reverse operation. If the **-R** flag is not specified, the **patch** command attempts to apply each portion in its reversed sense as well as in its normal sense, until a portion of the patch file is successfully applied. If the attempt is successful, the user is prompted to determine if the **-R** flag should be set.

> **Note:** This method cannot detect a reversed patch if used with a normal diff listing where the first command is an append (that is, would have been a delete). Appends always succeed because a null context matches anywhere. Fortunately, most patches add or change lines rather than delete lines. Therefore most reversed normal diff listings begin with a delete, causing a failure and triggering heuristics.

-s Patches silently unless an error occurs.

-v Prints the revision header and patch level. If the **-v** flag is used with other flags, the other flags are ignored.

-x *Number* Sets internal debugging flags. This flag is only for **patch** command developers.

Exit Status

The following exit values are returned:

0 Successful completion.

1 An error occurred.

Examples

1. To apply diff listings in the `difflisting` file to the `prog.c` file, enter:
   ```
   patch -i difflisting prog.c
   ```
2. To save the original version of the `prog.c` file, enter:
   ```
   patch -b -i difflisting prog.c
   ```
   This applies changes to `prog.c` and saves the original contents of `prog.c` in the file `prog.c.orig`.
3. To patch the `prog.c` file without altering the original version, enter:
   ```
   patch -i difflisting -o prog.new prog.c
   ```
   This uses `prog.c` as a source file, but the changed version is written to a file named `prog.new`.

Files

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

Related Information

The **diff** command, **ed** command.

---

pathchk Command

Purpose

Checks path names.

Syntax

pathchk [**-p**] *PathName* ...

Description

The **pathchk** command checks that one or more path names are valid and portable. By default, the **pathchk** command checks each component of each path name specified by the *path name* parameter based on the underlying file system. An error message is sent for each path name that meets the following criteria:

- The byte length of the full path name is longer than allowed by the system.
- The byte length of a component is longer than allowed by the system.
- Search permission is not allowed for a component.
- A character in any component is not valid in its containing directory.

It is not considered an error if one or more components of a path name do not exist, as long as a file matching the path name specified by the *PathName* parameter could be created that does not violate any of the above criteria.

More extensive portability checks are performed when the **-p** flag is specified.

Flags

-p Performs path name checks based on POSIX portability standards. An error message is sent for each path name that meets the following criteria:

- The byte length of the full path name is longer than allowed by POSIX standards.
- The byte length of a component is longer than allowed by POSIX standards.
- A character in any component is not in the portable file-name character set.

Exit Status

This command returns the following exit values:

0 All *PathName* operands passed all of the checks.
>0 An error occurred.

Examples

1. To check the validity and portability of the `/home/bob/work/tempfiles` path name on your system, enter:

   ```
   pathchk /home/bob/work/tempfiles
   ```

2. To check the validity and portability of the `/home/bob/temp` path name for POSIX standards, enter:

   ```
   pathchk -p /home/bob/temp
   ```

Files

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

Related Information

The **mhpath** command.

The File Systems in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices*.

---

pax Command

Purpose

Extracts, writes, and lists members of archive files; copies files and directory hierarchies.

Syntax

To List Member Files of Archived Files

pax [**-c**] [**-d**] [**-n**] [**-v**] [**-H** | **-L**] [**-f** *Archive*] [**-s** *ReplacementString* ...] [**-x** *Format*] [**-o** *Options*] [*Pattern* ...]

To Extract Archive Files Using the -r Flag

pax -r [**-c**] [**-d**] [**-i**] [**-k**] [**-n**] [**-u**] [**-v**] [**-H** | **-L**] [**-f** *Archive*] [**-o** *Options*] [**-p** *String* ...] [**-s** *ReplacementString* ...] [**-x** *Format*] [*Pattern* ...]

To Write Archive Files Using the -w Flag

pax -w [**-d**] [**-i**] [**-t**] [**-u**] [**-v**] [**-X**] [**-H** | **-L**] [**-b** *Blocking*] [[**-a**] **-f** *Archive*] [**-o** *Options*] [**-s** *ReplacementString* ...] [**-x** *Format*] [*File* ...]

To Copy Files Using the -r and -w Flags

pax -r -w [**-d**] [**-i**] [**-k**] [**-l**] [**-n**] [**-t**] [**-u**] [**-v**] [**-X**] [**-H** | **-L**] [**-p** *String* ...] [**-o** *Options*] [**-s** *ReplacementString* ...] [**-x** *Format*] [*File* ...] *Directory*

Description

The **pax** command extracts and writes member files of archive files; writes lists of the member files of archives; and copies directory hierarchies. The **-r** and **-w** flags specify the type of archive operation.

> **Note: pax** actively sparses files that are being restored. If a file has block aligned and sized areas that are NULL populated, **pax** does not cause physical space for those filesystem blocks to be allocated. The size in bytes of the file remains the same, but the actual space taken within the filesystem is only for the non-NULL areas.

Listing Member Files of Archived Files (List Mode)

When neither the **-r** nor the **-w** flags are specified, the **pax** command lists all the member files of the archive file read from standard input. If the *Pattern* parameter is specified, only the member files with pathnames that match the specified patterns are written to standard output. If a named file is a directory, the file hierarchy contained in the directory is also written. When neither the **-r** or **-w** flags are specified, the **-c**, **-d**, **-f**, **-n**, **-s**, and **-v** flags, and the *Pattern* parameter may be specified.

Extracting Archive Files Using the -r Flag (Read Mode)

When the **-r** flag is specified, but the **-w** flag is not, the **pax** command extracts all the member files of the archive files read from standard input. If the *Pattern* parameter is specified, only the member files with pathnames that match the specified patterns are written to standard output. If a named file is a directory, the file hierarchy contained in the directory is also extracted. The **-r** flag can be specified with the **-c**, **-d**, **-f**, **-i**, **-k**, **-n**, **-s**, **-u**, and **-v** flags, and with the *Pattern* parameter.

The access and modification times of the extracted files are the same as the archived files. The file modes of the extracted files are the same as when they were archived, unless they are affected by the user's default file creation mode (**umask**). The **S_ISUID** and **S_ISGID** bits of the extracted files are cleared.

If intermediate directories are necessary to extract an archive member, the **pax** command creates the directories with access permissions set as the bitwise inclusive OR of the values of the **S_IRWXU**, **S_IRWXG**, and **S_IRWXO** masks.

If the selected archive format supports the specification of linked files, it is an error if these files cannot be linked when the archive is extracted.

Writing Archive Files Using the -w Flag (Write Mode)

When the **-w** flag is specified and the **-r** flag is not, the **pax** command writes the contents of the files specified by the *File* parameter to standard output in an archive format. If no *File* parameter is specified, a list of files to copy, one per line, is read from the standard input. When the *File* parameter specifies a directory, all of the files contained in the directory are written. The **-w** flag can be specified with the **-a**, **-b**, **-d**, **-f**, **-i**, **-o**, **-s**, **-t**, **-u**, **-v**, **-x**, and **-X** flags and with *File* parameters.

When specifying pathnames that are greater than 100 characters for the United States Tape Archiver (USTAR) format, remember that the pathname is composed of a prefix buffer, a / (slash), and a name buffer. The prefix buffer can be a maximum of 155 characters and the name buffer can hold a maximum of 100 characters. If the pathname cannot be split into these two parts by a slash, it cannot be archived. This limitation is due to the structure of **tar** archive headers, and must be maintained for compliance with standards and backwards compatibility.

Copying Files Using the -r and -w Flags (Copy Mode)

When both the **-r** and **-w** flags are specified, the **pax** command copies the files specified by the *File* parameters to the destination directory specified by the *Directory* parameter. If no files are specified, a list of files to copy, one per line, is read from the standard input. If a specified file is a directory, the file hierarchy contained in the directory is also copied. The **-r** and **-w** flags can be specified with the **-d**, **-i**, **-k**, **-l**, **-o**, **-p**, **-n**, **-s**, **-t**, **-u**, **-v**, and **-X** flags and with *File* parameters. The *Directory* parameter must be specified.

Copied files are the same as if they were written to an archive file and subsequently extracted, except that there may be hard links between the original and the copied files.

Modifying the Archive Algorithm Using the -o Flag

Use the **-o** flag to modify the archive algorithm according to keyword-value pairs. The keyword-value pairs must adhere to a correct archive format. A list of valid keywords and their behavior is given in the subsequent description of the **-o** flag.

Further Notes

In read or copy modes, if intermediate directories are necessary to extract an archive member, **pax** performs actions equivalent to the **mkdir()** subroutine with the intermediate directory used as the path argument and the value **S_IRWXU** as the mode argument.

If any specified pattern or file operands are not matched by at least one file or archive member, **pax** writes a diagnostic message to standard error for each one that did not match and exits with an error status.

In traversing directories, **pax** will detect infinite loops; i.e., entering a previously visited directory that is an ancestor of the last file visited. Upon detection of an infinite loop, **pax** writes a diagnostic message to standard error and terminates.

When **pax** is in read mode or list mode, using the **-x pax** archive format, and a file name, link name, owner name, or any other field in an extended header record cannot be translated from the **pax** UTF8 codeset format to the current codeset and locale, **pax** writes a diagnostic message to standard error, processes the file as described for the **-o invalid=** option, and then processes the next file in the archive.

Variables

Directory Specifies the path of a destination directory when copying files.

File Specifies the path of a file to be copied or archived. If no file matches the *File* parameter, the **pax** command detects the error, exits, and writes a diagnostic message.

Pattern Specifies a pattern that matches one or more paths of archive members. A / (backslash) character is not recognized in the *Pattern* parameter and it prevents the subsequent character from having any special meaning. If no *Pattern* parameter is specified, all members are selected in the archive.

If a *Pattern* parameter is specified, but no archive members are found that match the pattern specified, the **pax** command detects the error, exits, and writes a diagnostic message.

Flags

-a Appends files to the end of an archive.

Note: Streaming tape devices do not allow append.

-b *Blocking* Specifies the block size for output. The *Blocking* parameter specifies a positive decimal integer value that specifies the number of bytes per block. Application conforming to POSIX2 should not specify a blocksize value greater than 32256. Devices and archive formats may impose restrictions on blocking. Blocking is automatically determined on input. Default blocking when creating archives depends on the archive format. (See the **-x** flag definition.)

The value of the *Blocking* parameter may be one of the following:

Integer **b**
: Specifies that the block size, in bytes, be the value of the positive decimal integer specified by the *Integer* parameter multiplied by 512.

Integer **k**
: Specifies that the block size, in bytes, be the value of the positive decimal integer specified by the *Integer* parameter multiplied by 1024.

Integer **m**
: Specifies that the block size, in bytes, be the value of the positive decimal integer specified by the *Integer* parameter multiplied by 1024 x 1024.

*Integer***+***Integer*
: Specifies that the block size, in bytes, be the sum of the positive decimal integers specified by the *Integer* parameters.

-c Matches all file or archive members except those specified by the *Pattern* parameter.

-d Causes directories being copied, archived, or extracted, to match only the directory itself and not the contents of the directory.

-f *Archive* Specifies the path of an archive file to be used instead of standard input (when the **-w** flag is not specified) or standard output (when the **-w** flag is specified but the **-r** flag is not). When specified with the **-a** flag option, any files written to the archive are appended to the end of the archive.

-H If a symbolic link referencing a directory is specified on the command line, **pax** archives the file hierarchy rooted in the directory referenced in the link, using the name of the link as the name of the file hierarchy. By default, **pax** archives the symbolic link itself.

-i Renames files or archives interactively. For each archive member that matches the *Pattern* parameter or file that matches a *File* parameter, a prompt is written to the display device that contains the name of a file or archive member. A line is then read from the display device. If this line is empty, the file or archive member is skipped. If this line consists of a single period, the file or archive member is processed with no modification to its name. Otherwise, its name is replaced with the contents of the line.

-k Prevents the **pax** command from writing over existing files.

-l Links files when copying files. Hard links are established between the source and destination file hierarchies whenever possible.

-L If a symbolic link referencing a directory is specified on the command line or encountered during the traversal of a file hierarchy, **pax** archives the file hierarchy rooted in the directory referenced in the link, using the name of the link as the name of the file hierarchy. By default, **pax** archives the symbolic link itself.

-n Selects the first archive member that matches each *Pattern* parameter. No more than one archive member is matched for each pattern.

-o *Options* Modifies the archiving algorithm according to the keyword-value pairs specified in the *Options* parameter. The keyword-value pairs must be in the following format:

keyword:=value,keyword:=value,...

Some keywords apply only to certain file formats, as indicated with each description. Use of keywords that are inapplicable to the file format being processed will be ignored by **pax**.

Keywords can be preceded with white space. The *value* field consists of zero or more characters; within *value*, any literal comma must be preceded with a backslash (\). A comma as the final character, or a comma followed solely by white space as the final character, in *Options* will be ignored. Multiple **-o** options can be specified. If keywords given to these multiple **-o** options conflict, the keywords and values appearing later in command-line sequences take precedence; earlier values are ignored.

The following keyword-value pairs are supported for the indicated file formats:

datastream=*pathname*,**datastr_size**=*size* (Applicable to all file formats.)

The **datastream** keyword indicates the incoming archive file is not in a file format; instead, it is a data stream from the standard input device. Consequently, the data should be archived as a regular file in a format recognized by the **-x** flag. The file name of the data stream should be specified in the *pathname* parameter and should include the identification of the person who invoked the command, the group identification, and the **umask** for the file mode.

> **Note:** The **datastream** keyword does not have a default variable size. You must specify one.

The **datastr_size** keyword denotes the size of the data stream input in bytes using decimal digits. If the **pax** command reaches the end of file (EOF) character before it reads the *size* parameter, it pads the archive file with null values. The null values make the archive file the same size as specified by the *size* parameter. If the data in the archive file exceeds the size specified, the **pax** command truncates the archive file to the size specified by the *size* parameter. The **pax** command also stops taking input and closes the archive file.

> **Note:** You can specify multiple instances of keyword pairs. If you assign different values to the same keyword, the **pax** command uses the last value assigned to the keyword to execute the **-o** flag.

delete=*pattern* (Applicable only to the **-x pax** format.)

When used in write or copy mode, **pax** omits any keywords matching *pattern* from the extended header records that it produces. When used in read or list mode, **pax** ignores any keywords matching *pattern* in the extended header records. In all cases, matching is done using standard shell pattern-matching notation. For example, `-o delete=security.*` suppresses security-related information.

exthdr.name=*string* (Applicable only to the **-x pax** format.)

This keyword allows user control over the name written into the **ustar** header blocks for the extended header records. The name is the contents of *string* after the following character substitutions have been made:

string **includes:** Replaced by:

%d The directory name of the file, equivalent to the result of the **dirname** utility on the translated pathname

%f The filename of the file, equivalent to the result of the **basename** utility on the translated pathname

%% A %% character

Any other % characters in *string* produce undefined results. If this keyword-value pair is not specified in the **-o** *Options* list, the default value of the name is:

%d/PaxHeaders/%f

globexthdr.name=*string* (Applicable only to the **-x pax** format.)

When used in write or copy mode with the appropriate options, **pax** creates global extended header records with **ustar** header blocks that will be treated as regular files by previous versions of **pax**. This keyword allows user control over the name that is written into the **ustar** header blocks for global extended header records. The name is the contents of *string* after the following character substitutions have been made:

string **includes:** Replaced by:

%n An integer that represents the sequence number of the global extended header record in the archive starting at 1

%% A % character

Any other % characters in *string* produce undefined results. If this keyword-value pair is not specified in the **-o** *Options* list, the default value of the name is

$TMPDIR/GlobalHead.%n

where **$TMPDIR** is either the value of the **TMPDIR** environment variable or **/tmp** if **TMPDIR** is unset.

invalid=*action* (Applicable only to the **-x pax** format.)

This keyword allows user control over the action **pax** takes upon encountering values in an extended header record that:

- in read or copy mode, are invalid in the destination hierarchy, or
- in list mode, cannot be written in the codeset and current locale.

-o *Options (Continued)*

pax recognizes these invalid values:

- In read or copy mode, a filename or link name that contains character encodings invalid in the destination hierarchy. (For example, the name may contain embedded NULLs.)
- In read or copy mode, a filename or link name that is longer than the maximum allowed in the destination hierarchy (for either a pathname component or the entire pathname).
- In list mode, any character string value (filename, link name, user name, etc.) that cannot be written in the codeset and current locale.

These mutually exclusive values of the *action* argument are supported:

- **bypass**

  In read or copy mode, **pax** bypasses the file, causing no change to the destination hierarchy. In list mode, **pax** writes all requested valid values for the file, but its method for writing invalid values is unspecified.
- **rename**

  In read or copy mode, **pax** acts as if the **-i** flag is in effect for each file with invalid filename or link name values, allowing the user to provide a replacement name interactively. In list mode, **pax** behaves identically to the **bypass** action.
- **UTF8**

  When used in read, copy, or list mode and a filename, link name, owner name, or any other field in an extended header record cannot be translated from the **pax UTF8** codeset format to the current codeset and locale, **pax** uses the actual UTF8 encoding for the name.
- **write**

  In read or copy mode, **pax** writes the file, translating or truncating the name, regardless of whether this may overwrite an existing file with a valid name. In list mode, **pax** behaves identically to the **bypass** action.

  If no **-o invalid=***action* is specified, **pax** acts as if the **bypass** action is specified. Any overwriting of existing files that may be allowed by the **-o invalid=***actions* is subject to permission (**-p**) and modification time (**-u**) restrictions, and is suppressed if the **-k** flag is also specified.

linkdata (Applicable only to the **-x pax** format.)

In write mode, **pax** writes the contents of a file to the archive, even when that file is merely a hard link to a file whose contents have already been written to the archive.

-o *Options (Continued)*

listopt=*format* (Applicable to all file formats.)

This keyword specifies the output format of the table of contents produced when the **-v** option is specified in list mode. To avoid ambiguity, this keyword-value pair should be used as the only or final keyword-value pair following the **-o** flag; all characters in the remainder of the option-argument are considered part of the format string. If multiple **-o listopt=format** options are specified, the format strings are considered to be a single, concatenated string, evaluated in command-line order. Please refer to the **List-Mode Format Specifications** section for more information.

times (Applicable only to the **-x pax** format.)

When used in write or copy mode, **pax** includes atime, ctime, and mtime extended header records for each file.

Extended header keywords

(Applicable only to the **-x pax** format.)

If the **-x pax** format is specified, any of the keywords and values defined in the list below can be used as parameters to the **-o** flag, in either of two modes:

keyword=value

When used in write or copy mode, these keyword-value pairs are written into the global extended header records of the new archive. When used in read or list mode, these keyword-value pairs act as if they were present in the global extended header records of the archive being read. In both cases, the given value is applied to all files that do not have a value assigned in their individual extended header records for the specified keyword.

*keyword***:=***value*

When used in write or copy mode, these keyword-value pairs are written into the extended header records of each file in the new archive. When used in read or list mode, these keyword-value pairs act as if they were present in the extended header records of each file in the archive being read. In both cases, the given value overrides any value for the specified keyword found in global or file-specific extended header records.

atime

The file access time for the following file(s), equivalent to the value of the st_atime member of the stat structure for a file.

charset

The name of the character set used to encode the data in the following file(s). The entries in this table are defined to refer to known standards:

value	Formal Standard
"ISO-IR 646 1990"	ISO/IEC 646 IRV
"ISO-IR 8859 1 1987"	ISO 8859-1
"ISO-IR 8859 2 1987"	ISO 8859-2
"ISO-IR 10646 1993"	ISO/IEC 10646
"ISO-IR 10646 1993 UTF8"	ISO/IEC 10646, UTF8 encoding
"BINARY"	None

The encoding is included in an extended header for information only; when **pax** is used as described, it does not translate the file data into any other encoding. The BINARY entry indicates binary data that is not encoded.

comment

A series of characters used as a comment. All characters in the value field are ignored by **pax**.

ctime

The file creation time for the following file(s), equivalent to the value of the st_ctime member of the stat structure for a file.

gid

The group ID of the group that owns the file, expressed as a decimal number using digits from ISO/IEC 646. This record overrides the *gid* field in the following header block(s). When used in write or copy mode, **pax** includes a gid extended header record for each file whose group ID is greater than 99,999,999.

gname

The group of the following file(s), formatted as a group name in the group database. This record overrides the *gid* and *gname* fields in the following header blocks, and any *gid* extended header record. When used in read, copy, or list mode, **pax** translates the name from the UTF8 encoding in the header record to the character set appropriate for the group database on the receiving system. If any of the UTF8 characters cannot be translated, and if the **-o invalid=UTF8** option is not specified, the results are undefined. When used in write or copy mode, **pax** includes a gname extended header record for each file whose group name cannot be represented entirely with the letters and digits of the portable character set.

linkpath

The pathname of a link being created to another file, of any type, previously archived. This record overrides the *linkname* field in the following **ustar** header block(s).

The following **ustar** header block determines the type of link created, whether hard or symbolic. In the latter case, the linkpath value is the contents of the symbolic link. **pax** translates the name of the link (contents of the symbolic link) from the UTF8 encoding to the character set appropriate for the local file system.

When used in write or copy mode, **pax** includes a linkpath extended header record for each link whose pathname cannot be represented entirely with the members of the portable character set other than NULL.

mtime

The file modification time of the following file(s), equivalent to the value of the st_mtime member of the stat structure for a file. This record overrides the *mtime* field in the following header block(s). The modification time is restored if the process has the appropriate privilege to do so.

path

The pathname of the following file(s). This record overrides the *name* and *prefix* fields in the following header block(s). **pax** translates the pathname of the file from the UTF8 encoding to the character set appropriate for the local file system. When used in write or copy mode, **pax** includes a path extended header record for each file whose pathname cannot be represented entirely with the members of the portable character set other than NULL.

realtime.*any*

The keywords prefixed by realtime are reserved for future POSIX realtime standardization. **pax** recognizes but silently ignores them.

security.*any*

The keywords prefixed by security are reserved for future POSIX security standardization. **pax** recognizes but silently ignores them.

size

The size of the file in octets, expressed as a decimal number using digits from ISO/IEC 646. This record overrides the *size* field in the following header block(s). When used in write or copy mode, **pax** includes a size of extended header record for each file with a size value greater than 999,999,999,999.

uid

The user ID of the user that owns the file, expressed as a decimal number using digits from ISO/IEC 646.. This record overrides the *uid* field in the following header block(s). When used in write or copy mode, **pax** includes a uid extended header record for each file whose owner ID is greater than 99,999,999.

uname

The owner of the following file(s), formatted as a user name in the user database. This record overrides the *uid* and *uname* fields in the following header block(s), and any *uid* extended header record. When used in read, copy, or list mode, **pax** translates the name from the UTF8 encoding in the header record to the character set appropriate for the user database on the receiving system. If any of the UTF8 characters cannot be translated, and if the **-o invalid=UTF8** option is not specified, the results are undefined. When used in write or copy mode, **pax** includes a uname extended header record for each file whose user name cannot be represented entirely with the letters and digits of the portable character set.

If the *value* field is zero length, it deletes any header block field, previously entered extended header value, or global extended header value of the same name.

If a keyword in an extended header record (or in a **-o** option-argument) overrides or deletes a corresponding field in the **ustar** header block, **pax** ignores the contents of that header block field.

Extended header keyword precedence

(Applicable only to the **-x pax** format.)

This section describes the precedence in which the various header records and fields and command-line options are selected to apply to a file in the archive. When **pax** is used in read or list modes, it determines a file attribute in this sequence:

1. If **-o delete**=*keyword-prefix* is used, the affected attribute is determined from step (7) if applicable, or ignored otherwise.
2. If **-o keyword**:=NULL is used, the affected attribute is ignored.
3. If **-o keyword**:=*value* is used, the affected attribute is assigned the value.
4. If *value* exists in a file-specific extended header record, the affected attribute is assigned the value. When extended header records conflict, the last one given in the header takes precedence.
5. If **-o keyword**=*value* is used, the affected attribute is assigned the value.
6. If a value exists in a global extended header record, the affected attribute is assigned the value. When global extended header records conflict, the last one given in the global header takes precedence.
7. Otherwise, the attribute is determined from the **ustar** header block.

-p *String*

Specifies one or more file characteristics to be retained or discarded on extraction. The *String* parameter consists of the characters **a**, **e**, **m**, **o**, and **p**. Multiple characteristics can be concatenated within the same string and multiple **-p** flags can be specified. The specifications have the following meanings:

a Does not retain file-access times.

e Retains the user ID, group ID, file mode, access time, and modification time.

m Does not retain file-modification times.

o Retains the user ID and the group ID.

p Retains the file modes.

If neither the **-e** nor the **-o** flag is specified, or the user ID and group ID are not preserved for any reason, the **pax** command does not set the **S_ISUID** and **S_ISGID** bits of the file mode. If the retention of any of these items fails, the **pax** command writes a diagnostic message to standard error. Failure to retain any of the items affects the exit status, but does not cause the extracted file to be deleted. If specification flags are duplicated or conflict with each other, the last flag specified takes precedence. For example, if `-p eme` is specified, file-modification times are retained.

-r

Reads an archive file from the standard input.

-s *ReplacementString*

Modifies file- or archive-member names specified by the *Pattern* or *File* parameters according to the substitution expression *ReplacementString*, using the syntax of the **ed** command. The substitution expression has the following format:

-s /*old*/*new*/[gp]

where (as in the **ed** command), *old* is a basic regular expression and *new* can contain an **&** (ampersand), **\n** (**n** is a digit) back references, or subexpression matching. The *old* string can also contain new-line characters.

Any non-null character can be used as a delimiter (the / (backslash) is the delimiter in the example). Multiple **-s** flag expressions can be specified; the expressions are applied in the order specified, terminating with the first successful substitution. The optional trailing `g` character performs as in the **ed** command. The optional trailing `p` character causes successful substitutions to be written to standard error. File or archive-member names that substitute to the empty string are ignored when reading and writing archives.

-t

Causes the access times of input files to be the same as they were before being read by the **pax** command.

-u

Ignores files that are older than a preexisting file or archive member with the same name.

- When extracting files, an archive member with the same name as a file in the file system is extracted if the archive member is newer than the file.
- When writing files to an archive file, an archive member with the same name as a file in the file system is superseded if the file is newer than the archive member. If the **-a** flag is specified this is accomplished by appending to the archive. Otherwise it is unspecified if this is accomplished by actual replacement in the archive or by appending to the archive.
- When copying files to a destination directory, the file in the destination hierarchy is replaced by the file in the source hierarchy or by a link to the file in the source hierarchy if the file in the source hierarchy is newer.

-v

Writes information about the process. If neither the **-r** or **-w** flags are specified, the **-v** flag produces a verbose table of contents; otherwise, archive member pathnames are written to standard error.

-w

Writes files to the standard output in the specified archive format.

-x *Format*

Specifies the output archive format. The **pax** command recognizes the following formats:

pax The default interchange format. The default blocking value for this format for character-special archive files is 10240. Blocking values from 512 to 32256 in increments of 512 are supported.

cpio Extended **cpio** interchange format. The default blocking value for this format for character-special archive files is 5120. Blocking values from 512 to 32256 in increments of 512 are supported.

ustar Extended **tar** interchange format. The default blocking value for this format for character-special archive files is 10240. Blocking values from 512 to 32256 in increments of 512 are supported.

Any attempt to append to an archive file in a format different from the existing archive format causes the **pax** command to exit immediately with a nonzero exit status.

In copy mode, if no **-x** format is specified, **pax** behaves as if **-x pax** were specified.

-X When traversing the file hierarchy specified by a pathname, the **pax** command does not descend into directories that have a different device ID.

Flag Interaction and Processing Order

The flags that operate on the names of files or archive members (**-c**, **-i**, **-n**, **-s**, **-u**, and **-v**) interact as follows:

- When extracting files, archive members are selected according to the user-specified *pattern* parameters as modified by the **-c**, **-n**, and **-u** flags. Then, any **-s**, and **-i** flags modify, in that order, the names of the selected files. The **-v** flag writes the names resulting from these modifications.
- When writing files to an archive file, or when copying files, the files are selected according to the user-specified pathnames as modified by the **-n** and **-u** flags. Then, any **-s**, and **-i** flags modify, in that order, the names resulting from these modifications. The **-v** flag writes the names resulting from the modification.
- If both the **-u** and **-n** flags are specified, the **pax** command does not consider a file selected unless it is newer than the file to which it is compared.

List Mode Format Specifications

In list mode with the **-o listopt=***format* option, the format argument is applied for each selected file. **pax** appends a newline character to the **listopt** output for each selected file. The format argument is used as the format string described in **printf()**, with the following exceptions:

1. The sequence *keyword* can occur before a format conversion specifier. The conversion argument is defined by the value of *keyword*. The following keywords are supported:
   - Any of the field name entries for **ustar** and **cpio** header blocks.
   - Any keyword defined for the extended header or provided as an extension within the extended header.

   For example, the sequence `%(charset)s` is the string value of the name of the character set in the extended header.

   The result of the keyword conversion argument is the value from the applicable header field or extended header, without any trailing NULLs.

   All keyword-values used as conversion arguments are translated from the UTF8 encoding to the character set appropriate for the local file system, user database, etc., as applicable.

2. An additional conversion character, **T**, specifies time formats. The **T** conversion character can be preceded by the sequence *keyword=subformat*, where *subformat* is a date format allowed by the **date** command. The default keyword is **mtime** and the default subformat is: %b %e %H:%M %Y.
3. An additional conversion character, **M**, specifies the file mode string as displayed by the **ls -l** command. If *keyword* is omitted, the **mode** keyword is used. For example, %.1M writes the single character corresponding to the *entry type* field of the **ls -l** command.
4. An additional conversion character, **D**, specifies the device for block or special files, if applicable. If not applicable and *keyword* is specified, then this conversion is equivalent to %*keyword* u. If not applicable and *keyword* is omitted, this conversion is equivalent to <space>.
5. An additional conversion character, **F**, specifies a pathname. The **F** conversion character can be preceded by a sequence of comma-separated keywords:

   keyword,keyword...

   The values for all the non-null keywords are concatenated together, each separated by a /. The default is *path* if the keyword path is defined; otherwise, the default is *prefix,name*.
6. An additional conversion character, **L**, specifies a symbolic link expansion. If the current file is a symbolic link, then %L expands to:

   "%s -> %s", *value_of_keyword*, *contents_of_link*

   Otherwise, the %L conversion character is equivalent to %F.

Exit Status

This command returns the following exit values:

0 Successful completion.
>0 An error occurred.

Examples

1. To copy the `olddir` directory hierarchy to `newdir`, enter:

   ```
   mkdir newdir
   pax -rw olddir newdir
   ```
2. To copy the contents of the current directory to the tape drive, enter:

   ```
   pax -wf /dev/rmt0
   ```
3. To archive the file `xxx` as `XXX` and display the successful substitution, enter:

   ```
   pax -wvf/dev/rfd0 -s /xxx/XXX/p xxx
   ```

   OR

   ```
   pax -wvf/dev/rfd0 -s/x/X/gp xxx
   ```
4. To read a file from a standard input and dump it to a datastream file with a specified size, enter:

   ```
   dd if=/dev/hd6 bs=36b count=480 | pax -wf /dev/rfd0 -o
   datastream=_filename_,datastr_size=_size_
   ```

Files

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

Related Information

The **ed** command, **cpio** command, **tar** command, and **rmt** command.

The Files Overview in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices*.

pcat Command

Purpose

Unpacks files and writes them to standard output.

Syntax

pcat *File* ...

Description

The **pcat** command reads the files designated by the *File* parameter, unpacks them, and writes them to standard output. Whether or not the specified file ends in the **.z** characters, the **pcat** command assumes that the file is packed and unpacks it.

The exit value of the **pcat** command is the number of files it was unable to unpack. A file cannot be unpacked if any of the following occurs:

- The file name (exclusive of **.z**) has more than 253 bytes.
- The file cannot be opened.
- The file is not a packed file.

Exit Status

This command returns the following exit values:

0 Successful completion.
>0 An error occurred.

Examples

1. To display compressed files, enter:

   ```
   pcat chap1.z chap2 | pg
   ```

   This command sequence displays the compressed files `chap1.z` and `chap2.z` on the screen in expanded form, a page at a time (`| pg`). Note that the **pcat** command accepts files with and without the `.z` characters.
2. To use a compressed file without expanding the copy stored on disk, enter:

   ```
   pcat chap1.z | grep 'Greece'
   ```

   This command sequence prevents the **pcat** command from displaying the contents of `chap1.z` in its expanded form and pipes it to the **grep** command.

File

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

Related Information

The **cat** command, **grep** command, **pack** command, **unpack** command.

Files Overview.and Input and Output Redirection Overview in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices*.

pdelay Command

Purpose

Enables or reports the availability of delayed login ports.

Syntax

pdelay [**-a**] [*Device*]

Description

The **pdelay** command enables delayed ports. Delayed ports are enabled like shared ports, except that the login herald is not displayed until you type one or more characters (usually carriage returns). If a port is directly connected to a remote system or connected to an intelligent modem, it is enabled as a delayed port to prevent the **getty** command from talking to a **getty** on the remote side or to the modem on a local connection. This action conserves system resources and is equivalent to **pdelay enabled=delay**. If you do not specify a *Device* parameter, the **pdelay** command reports the names of the currently enabled ports.

Use the *Device* parameter to specify the ports to be enabled. Permitted values include:

- Full device name, such as the **/dev/tty1** device
- Simple device name, such as the **tty1** device
- A number (for example, 1 to indicate the **/dev/tty1** device)

Note: You must have root user authority to run this command.

Flags

-a Enables all ports as delayed.

Example

To display the names of the delayed ports that are currently enabled, enter:

```
pdelay
```

Files

/etc/locks	Contains **lock** files for the **pshare** and **pdelay** commands.
/usr/sbin/pdelay	Contains the **pdelay** command.

Related Information

The **getty** command, **init** command, **pdisable** command, **penable** command, **phold** command, **pshare** command, **pstart** command.

pdisable Command

Purpose

Disables login ports.

Syntax

pdisable [**-a**] [*Device*]

Description

The **pdisable** command disables a specific port, even if a user is logged in at that port. The system disables a port by updating an entry in the **/etc/inittab** file and then sending a signal to the **init** process**.** When the **init** process receives the signal and reads the updated status entry, it takes the appropriate action.

Use the *Device* parameter to specify the ports to be disabled. Permitted values include:

- A full device name, such as the **/dev/tty1** device
- A simple device name, such as the **tty1** device
- A number (for example, 1 to indicate the **/dev/tty1** device).

If you do not specify a *Device* parameter, the **pdisable** command reports the names of currently disabled ports in its set.

Note: You must have root user authority to run this command.

Flag

-a Disables all ports that are currently enabled.

Examples

1. To display the names of all ports currently disabled, enter:
   ```
   pdisable
   ```
2. To disable all ports that are enabled, even if users are logged in, enter:
   ```
   pdisable -a
   ```
3. To disable the workstation attached to the **/dev/tty8** port, enter:
   ```
   pdisable tty8
   ```

Files

/etc/locks	Contains **lock** files for the **pshare** and **delay** commands.
/usr/sbin/pdisable	Contains the **pdisable** command.

Related Information

The **init** command, **pdelay** command, **penable** command, **phold** command, **pshare** command, **pstart** command.

The **inittab** file.

---

penable Command

Purpose

Enables or reports the availability of login ports.

Syntax

penable [**-a**] [*Device*]

Description

The **penable** command enables normal ports. Normal ports are asynchronous and only allow users to log in. No outgoing use of the port is allowed while it is enabled. The system enables a port by updating an entry in the **/etc/inittab** file and then sending a signal to the **init** process. After receiving the signal and reading the updated status entry, the process takes the appropriate action.

Use the *Device* parameter to specify the ports to be enabled. Permitted values include:

- Full device name, such as the **/dev/tty1** device
- Simple device name, such as the **tty1** device
- A number (for example, 1 to indicate the **/dev/tty1** device).

If you do not specify a *Device* parameter, the **penable** command reports the names of the currently enabled normal ports.

Note: You must have root user authority to run this command.

Flags

-a Enables all normal ports.

Example

To enable all normal ports listed in the **/etc/inittab** file, enter:

```
penable -a
```

Files

/etc/locks	Contains **lock** files for the **pshare** and **pdelay** commands.
/usr/sbin/penable	Contains the **penable** command.

Related Information

The **init** command, **pdelay** command, **pdisable** command, **phold** command, **pshare** command, **pstart** command.

The **inittab** file.

---

pg Command

Purpose

Formats files to the display.

Syntax

pg [**-** *Number*] [**-c**] [**-e**] [**-f**] [**-n**] [**-p** *String*] [**-r**] [**-s**] [**+***LineNumber*] [**+**/*Pattern*/] [*File ...*]

Description

The **pg** command reads a file name from the *File* parameter and writes the file to standard output one screen at a time. If you specify a **-** (dash) as the *File* parameter, or run the **pg** command without options, the **pg** command reads standard input. Each screen is followed by a prompt. If you press the Enter key, another page is displayed. Subcommands used with the **pg** command let you review or search in the file.

To determine workstation attributes, the **pg** command scans the file for the workstation type specified by the **TERM** environment variable. The default type is **dumb**.

When the **pg** command pauses and issues a prompt, you can issue a subcommand. Some of these subcommands change the display to a particular place in the file, some search for specific patterns in the text, and others change the environment in which the **pg** command works.

Changing Location Within the File

The following subcommands display a selected place in the file:

Page	Displays the page specified by the *Page* parameter.
+*Number*	Displays the page obtained by adding the *Number* value to the current page.
-*Number*	Displays the page as specified by the *Number* value before the current page.
l	(Lowercase L) Scrolls the display one line forward.
*Number***l**	Displays at the top of the screen the line specified by the *Number* parameter.
+*Number***l**	Scrolls the display forward for the specified number of lines.
-*Number***l**	Scrolls the display backward for the specified number of lines.
d	Scrolls half a screen forward. Pressing the Ctrl-D key sequence functions the same as the **d** subcommand.
-d	Scrolls half a screen backward. Pressing the **-**Ctrl-D key sequence functions the same as the **-d** subcommand.
Ctrl-L	Displays the current page again. A single . (dot) functions the same as the **Ctrl-L** key sequence subcommand.
$	Displays the last page in the file. Do not use this when the input is from a pipeline.

Searching for Text Patterns

The following subcommands search for text patterns in the text. (You can also use the patterns described in the **ed** command.) They must always end with a new-line character, even if the **-n** flag is used.

In an expression such as `[k.a-z]k.`, the minus implies a range, as in a through z, according to the current collating sequence. A collating sequence defines equivalence classes for use in character ranges.

[*Number***]/***Pattern*/

Searches for the occurrence of the *Pattern* value as specified by the *Number* variable. The search begins immediately after the current page and continues to the end of the current file, without wraparound. The default for the *Number* variable is 1.

*Number***?***Pattern***?**

*Number***^***Pattern***^**

Searches backward for the occurrence of the *Pattern* value as specified by the *Number* variable. The searching begins immediately before the current page and continues to the beginning of the current file, without wraparound. The default for the *Number* variable is 1. The ^ notation is useful for Adds 100 terminals which will not properly handle the ? notation.

After searching, the **pg** command displays the line with the matching pattern at the top of the screen. You can change the position of the display by adding the **m** or **b** suffix to the search command. The **m** suffix displays the line with the matching pattern in the middle of the screen for all succeeding subcommands. The **b** suffix displays the line with the matching pattern at the bottom of the screen for all succeeding subcommands. The **t** suffix displays the line with the matching pattern at the top of the screen again.

Changing the pg Environment

You can change the **pg** command environment with the following subcommands:

[*Number***]n**

Begins examining the next file in the command line, as specified by the *Number* variable. The default for the *Number* variable is first.

[*Number***]p**	Begins examining the previous file on the command line, as specified by the *Number* variable. The default for the *Number* variable is first.
[*Number***]w**	Displays another window of text. If the *Number* variable is specified, sets the window size to the number of lines it specifies. This subcommand is the same as the **[***Number***]z** subcommand.
[*Number***]z**	Displays another window of text. If the *Number* variable is specified, sets the window size to the number of lines it specifies. This subcommand is the same as the **[***Number***]w** subcommand.
s *File*	Saves the input in the specified file. Only the current file being examined is saved. This command must always end with a new-line character, even if you specify the **-n** flag.
h	Displays an abbreviated summary of available subcommands.
q or **Q**	Quits the **pg** command.
!*Command*	Sends the specified command to the shell named in the **SHELL** environment variable. If this is not available, the default shell is used. This command must always end with a new-line character, even if the **-n** flag is used.

Attention:

1. Some output is lost when you press the QUIT WITH DUMP (Ctrl-\) or INTERRUPT (Ctrl-C) key sequence because any characters waiting in the output queue are purged when the **QUIT** signal is received.
2. If workstation tabs are not set every eight positions, unpredictable results can occur.

At any time output is being sent to the workstation, you can press the QUIT WITH DUMP or INTERRUPT key sequence. This causes the **pg** command to stop sending output and displays the prompt. Then you can enter one of the preceding subcommands at the command prompt.

If standard output is not a workstation, the **pg** command acts like the **cat** command, except that a header is displayed before each file.

While waiting for workstation input, the **pg** command stops running when you press the INTERRUPT key sequence. Between prompts these signals interrupt the current task and place you in the prompt mode.

Flags

-c	Moves the cursor to the home position and clears the screen before each page. This flag is ignored if the `clear_screen` field is not defined for your workstation type in the **terminfo** file.
-e	Does not pause at the end of each file.
-f	Does not split lines. Normally, the **pg** command splits lines longer than the screen width.
-n	Stops processing when a **pg** command letter is entered. Normally, commands must end with a new-line character.
-p *String*	Uses the specified string as the prompt. If the *String* contains a **%d** value, that value is replaced by the current page number in the prompt. The default prompt is **:** (colon). If the specified string contains spaces, you must enclose the string in quotation marks.
-r	Prevents shell escape when the ″**!**″ subcommand is used.
-s	Highlights all messages and prompts.
+*LineNumber*	Starts at the specified line number.
-*Number*	Specifies the number of lines in the window. On workstations that contain 24 lines, the default is 23.
+/*Pattern***/**	Starts at the first line that contains the specified pattern.

Exit Status

This command returns the following exit values:

0	Successful completion.
>0	An error occurred.

Example

To look at the contents of a file one page at a time, enter:

```
pg filename
```

Files

/usr/bin/pg	Contains the **pg** command.
/usr/share/lib/terminfo/*	Contains the **terminfo** file that defines terminal types.
/tmp/pg*	Contains the temporary file created when using **pg** command.

Related Information

The **cat** command, **ed** command, **grep** command.

File and Directory Access Modes in the *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*.

Input and Output Redirection Overview and Shells Overview in the *AIX 5L Version 5.2 System User's Guide: Operating System and Devices*.

Files Overview in the *AIX 5L Version 5.2 System User's Guide: Operating System and Devices*.

---

phold Command

Purpose

Disables or reports the availability of login ports on hold.

Syntax

phold [**-a**] [*Device*]

Description

The **phold** command disables a set of login ports. The **phold** command allows logged-in users to continue, but does not allow any more users to log in. A user cannot log in on a disabled port. The system disables a port by updating an entry in the **/etc/inittab** file and then sending a signal to the **init** process. When the **init** process receives the signal and reads the updated status entry, it takes the appropriate action.

Use the *Device* parameter to specify the ports to be disabled. Permitted values include:

- A full device name, such as the **/dev/tty1** device
- A simple device name, such as the **tty1** device
- A number (*e.g.*, 1 to indicate the **/dev/tty1** device)

If you do not specify a *Device* parameter, the **phold** command reports the names of currently disabled ports in its set.

> **Note:** You must have root user authority to run this command.

Flags

-a Holds all ports that are currently enabled.

Example

To list the ports that are currently on hold, enter:

```
phold
```

Files

/etc/locks	Contains **lock** files for the **pshare** and **pdelay** commands.
/etc/phold	Contains the **phold** command.

Related Information

The **init** command, **pdelay** command, **pdisable** command, **penable** command, **pshare** command, **pstart** command.

The **inittab** file.

---

pic Command

Purpose

Preprocesses **troff** command input for the purpose of drawing pictures.

Syntax

pic [**-T** *Name*] [**-** | *File* ...]

Description

The **pic** command is a **troff** command preprocessor for drawing simple figures on a typesetter. The basic objects are a box, circle, ellipse, line, spline, arrow, arc, and the text specified by the *Text* variable. The top-level object is the picture.

File Specifies the output from a **troff** command that is processed by the **pic** command to draw pictures.

Pictures

The top-level object in the **pic** command is the picture.

.PS *OptionalWidth OptionalHeight*

ElementList

.PE

If the **.PF** macro is used instead of the **.PE** macro, the position after printing is restored to what it was upon entry.

OptionalWidth	Specifies the width of the picture (in inches), if present, regardless of any dimensions used internally. The maximum value is 8.5.
OptionalHeight	Specifies a height value, in inches, different from the default, which is scaled to the same proportion. The maximum value is 14.

ElementList Represents the following list of elements:

Shape AttributeList
For Statement

Placename: Element
If Statement

Placename: Position
Copy Statement

Variable = Expression
Print Statement

Direction
Plot Statement

{ *List of Elements* **}**
sh *X Commandline X*

[*List of Elements* **]**
troff-command

Variable names begin with a lowercase letter, followed by zero or more letters or numbers. Place names begin with an uppercase letter, followed by zero or more letters or numbers. Place and variable names retain their values from one picture to the next.

Elements in a list must be separated by new-line characters or ; (semicolon); a long element can be continued by ending the line with a \ (backslash). Comments are introduced by a # character and ended by a new-line character.

Primitives

The primitive objects are as follows:

box

circle

ellipse

arc

line

arrow

spline

move

Text-List

The **arrow** object is the same as the **line** object with the **->** attribute.

Attributes

An *AttributeList* element is a sequence of zero or more attributes; each attribute consists of a keyword, perhaps followed by a value.

Attribute	**Attribute**
h(eigh)t *Expression*	**wid(th)** *Expression*
rad(ius) *Expression*	**diam(eter)** *Expression*
up *OptionalExpression*	**down** *OptionalExpression*
right *OptionalExpression*	**left** *OptionalExpression*
from *Position*	**to** *Position*

Attribute	Attribute
at *Position*	**with** *Corner*
by *Expression, Expression*	**then**
dotted *OptionalExpression*	**dashed** *OptionalExpression*
chop *OptionalExpression*	**-> <- <->**
invis	**same**
Text-list	

Missing attributes and values are filled in from defaults. Not all attributes make sense for all primitives; irrelevant ones are not processed. The following are the currently meaningful attributes:

Primitives	Attributes
box	**h(eigh)t**, **wid(th)**, **at**, **same**, **dotted**, **dashed**, **invis**, *Text*
circle, **ellipse**	**rad(ius)**, **diam(eter)**, **h(eigh)t**, **wid(th)**, **at**, **same**, **invis**, *Text*
arc	**up**, **down**, **left**, **right**, **h(eigh)t**, **wid(th)**, **from**, **to**, **at**, **rad(ius)**, **invis**, **ccw**, **cw**, **<-**, **->**, **<->**, *Text*
line, **arrow**	**up**, **down**, **left**, **right**, **h(eigh)t**, **wid(th)**, **from**, **to**, **by**, **then**, **at**, **same**, **dotted**, **dashed**, **invis**, **<-**, **->**, **<->**, *Text*
spline	**up**, **down**, **left**, **right**, **h(eigh)t**, **wid(th)**, **from**, **to**, **by**, **then**, **at**, **same**, **invis**, **<-**, **->**, **<->**, *Text*
move	**up**, **down**, **left**, **right**, **to**, **by**, **same**, *Text*
Text-list	**at**, *Text-item*

The **at** attribute implies placing the geometrical center at the specified place. For lines, splines, and arcs, the **h(eigh)t** and **wid(th)** attributes refer to arrowhead size.

The *Text-item* variable is usually an attribute of some primitive; by default, it is placed at the geometrical center of the object. Stand-alone text is also permitted. A *Text-list* primitive is a list of text items; a text item is a quoted string optionally followed by a positioning request, as follows:

″...″

″...″ **center**

″...″ **ljust**

″...″ **rjust**

″...″ **above**

″...″ **below**

If there are multiple text items for some primitives, they are centered vertically except as qualified. Positioning requests apply to each item independently.

Text items can contain **troff** commands that control, for example, size and font changes and local motions. Make sure these commands are balanced so that the entering state is restored before exiting.

Positions/Places

A position is ultimately an *X,Y* coordinate pair, but it can also be specified in the following ways:

Place

(*Position*)

Expression, Expression

(*Position*) [**+/-** (*Expression, Expression*)]

(*Position*) [**+/-** *Expression, Expression*]

(*Place1, Place2*)

(*Place1.X, Place2.Y*)

Expression **<** *Position, Position* **>**

Expression [**of the way**] **between** *Position* **and** *Position*

Placename [*Corner*]

Corner Placename

Here

Corner **of** *Nth Shape*

Nth shape [*Corner*]
Note: A *Corner* variable designates one of the eight compass points or the center, beginning, or end of a primitive, as follows:

.n .e .w .s .ne .se .nw .sw

.t .b .r .l

c .start .end

Each object in a picture has an ordinal number; *Nth* refers to this, as follows:

- *Nth*
- *Nth* last

The **pic** command is flexible enough to accept names like **1***th* and **3***th*. Usage like **1***st* and **3***st* are accepted as well.

Variables

The built-in variables and their default values are as follows:

boxwid	0.75
boxht	0.5
circlerad	0.25
arcrad	0.25
ellipsewid	0.75
ellipseht	0.5
linewid	0.5
lineht	0.5
movewid	0.5
moveht	0.5
arrowwid	0.05

arrowht	0.1
textwid	0
textht	0
dashwid	0.5
scale	1

These default values can be changed at any time, and the new values remain in force from picture to picture until changed again.

The **textht** and **textwid** variables can be set to any value to control positioning. The width and height of the generated picture can be set independently from the **.PS** macro line. Variables changed within the [(left bracket) delimiter and the] (right bracket) delimiter revert to their previous value upon exit from the block. Dimensions are divided by **scale** during output.

Note: The **pic** command has an eight inch by eight inch limitation on picture sizes generated and sent to the **troff** command, even when the **.ps** (size) line specifies a size greater than eight inches.

Expressions

The following **pic** command expressions are evaluated in floating point. All numbers representing dimensions are taken to be in inches.

Expression + *Expression*

Expression - *Expression*

Expression * *Expression*

Expression / *Expression*

Expression % *Expression* (modulus)

- *Expression*

(*Expression*)

variable

number

Place **.x**

Place **.y**

Place **.ht**

Place **.wid**

Place **.rad**

sin(*Expression*) **cos**(*Expression*) **atan2**(*Expression, Expression*) **log**(*Expression*) **sqrt**(*Expression*)
int(*Expression*) **max**(*Expression, Expression*) **min**(*Expression,Expression*) **rand**(*Expression*)

Logical Operators

The **pic** command provides the following operators for logical evaluation:

!	Not
>	Greater than
<	Less than
>/=	Greater than or equal to
</=	Less than or equal to
&&	And
I	Or
==	Equal to
!=	Not equal to

Definitions

The following **define** statement is not part of the grammar:

define *Name* **X** *Replacement text* **X**

Occurrences of values such as **$1** and **$2** in the *Replacement text* variable are replaced by the corresponding options if the *Name* variable is called, as follows:

Name(*Option1, Option2*, ...)

Non-existent options are replaced by null strings. The *Replacement text* variable can contain newline characters.

copy and copy thru Statements

The **copy** statement includes data from a file or values that immediately follow, such as:

```
copy File
copy thru Macro
copy File thru Macro
copy File thru Macro until String
```

The *Macro* parameter value can be either the name of a defined macro or the body of a macro enclosed in some character not part of the body. If no file name is given, the **copy** statement copies the input until the next **.PE** macro line.

for Loops and if Statements

The **for** and **if** statements provide for loops and decision-making, as follows:

```
Variable=Expression to Expression by Expression do X anything X
if Expression then X anything X else X anything X
```

The **by** and **else** clauses are optional. The *Expression* variable in an **if** statement can use the usual relational operators or the *String1* **==** (or **!=**) *String2* string tests.

Miscellaneous Information

The **sh** command runs a command line, as follows:

```
sh X Commandline X
```

It is possible to plot the value of an expression, as follows:

```
plot Expression OptionalFormat Attributes
```

The *Expression* variable value is evaluated and converted to a string (using the format specification, if provided).

The state of fill or no-fill mode is preserved with respect to pictures.

Input numbers can be expressed in **E** (exponential) notation.

Flags

-T*Name* Prepares the output for the specified printing device. Possible values for *Name* are:

ibm3812
3812 Pageprinter.

ibm3816
3816 Pageprinter.

hplj Hewlett-Packard LaserJet II.

ibm5587G
5587-G01 Kanji Printer multi-byte language support.

psc PostScript printer.

X100 AIXwindows display.

X100K AIXwindows display for multi-byte character support.

The default is **ibm3816**.
Note: It is possible to set the **TYPESETTER** environment variable to one of the preceding values instead of using the **-T***Name* flag of the **troff** command.

- Reverts to standard input.

Related Information

The **grap** command, **sh** command,**troff** command.

The **.PE** macro, **.PF** macro, **.PS** macro, **me** macro package.

---

pick Command

Purpose

Selects messages by content and creates and modifies sequences.

Syntax

pick [**+***Folder*] [*Messages*] [**-datefield** *Field*] [**-not**] [**-lbrace**] [**-after** *Date*] [**-before** *Date*] [**-cc** *"Pattern"*] [**-date** *"Pattern"*] [**-from** *"Pattern"*] [**-search** *"Pattern"*] [**-to***"Pattern"*] [**—***Component "Pattern"*] [**-rbrace**] [**-and**] [**-or**] [**-sequence** *Name* [**-zero** | **-nozero**] [**-public** | **-nopublic**] [**-list** | **-nolist**]

Description

The **pick** command selects messages containing particular character patterns or particular dates. You can use the **-and**, **-or**, **-not**, **-lbrace**, and **-rbrace** flags to construct compound conditions for selecting messages.

Flags

-after *Date*
Selects messages with dates later than that specified by the *Date* variable. Use the following specifications for the *Date* variable:

yesterday	**today**	**tomorrow**
sunday	**monday**	**tuesday**
wednesday	**thursday**	**friday**
saturday	**-**Days	SystemDate

The **pick** command treats the days of the week as days in the past. For example, **monday** means last Monday, not today or next Monday. You can use the **-***Days* argument to specify a number of days in the past. For example, -31 means 31 days ago. For the *SystemDate* argument, you can specify any valid format defined for your system.

-and
Forms a logical AND operation between two message-selecting flags; for example, `pick -after Sunday -and -from mark`. The **-and** flag has precedence over the **-or** flag, but the **-not** flag has precedence over the **-and** flag. Use the **-lbrace** and **-rbrace** flags to override this precedence.

-before *Date*
Selects messages with dates earlier than the specified date. See the **-after** flag on how to specify *Date*.

-cc *"Pattern"*
Selects messages that contain the character string specified by the *"Pattern"* variable in the `cc:` field.

-date *"Pattern"*
Selects messages that contain the character string specified by the *"Pattern"* variable in the `Date:` field.

-datefield *Field*
Specifies which dated field is parsed when the **-after** and **-before** flags are given. By default, the **pick** command uses the `Date:` field.

+*Folder*
Identifies the folder that contains the messages you wish to pick. By default, the system uses the current folder.

-from *"Pattern"*
Selects messages that contain the character string specified by the *"Pattern"* variable in the `From:` field.

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

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

-lbrace
Groups **-and**, **-or**, and **-not** operations. Operations between the **-lbrace** and **-rbrace** flags are evaluated as one operation. You can nest the **-lbrace** and **-rbrace** flags.

-list
Sends a list of selected message numbers to standard output. This allows you to use the **pick** command to generate message numbers to use as input for other commands. For example, to scan all messages in the current folder that were sent after Tuesday, you would enter the following:

```
scan 'pick -after tuesday -list'
```

If you do not specify a sequence, the **-list** flag is the default.

Messages
Specifies the messages to search. You can specify several messages, a range of messages, or a single message. Use the following to specify messages:

Number
Number of the message.

Sequence
A group of messages specified by the user. Recognized values include:

all All of the messages in the folder. This is the default.

cur or . (period)
Current message.

first First message in a folder.

last Last message in a folder.

new New message that is created.

next Message following the current message.

prev Message preceding the current message.

-nolist
Prevents the **pick** command from generating a list of the selected message numbers. If a sequence is specified, the **-nolist** flag is the default.

-nopublic
Restricts a sequence to your usage. The **-nopublic** flag does not restrict the messages in a sequence, only the sequence itself. This option is the default if the folder is write-protected from other users.

-not
Forms a logical NOT operation on a message-selecting flag; for example, `pick -not -from george`. This construction evaluates all messages not chosen by the message-selecting flag. The **-not** flag has precedence over the **-and** flag, and the **-and** flag has precedence over the **-or** flag. Use the **-lbrace** and **-rbrace** flags to override this precedence.

-nozero
Appends the selected messages to the specified sequence.

-or
Forms a logical OR operation on two message-selecting flags; for example, `pick -from amy -or -from mark`. The **-not** flag has precedence over the **-and** flag, and the **-and** flag has precedence over the **-or** flag. Use the **-lbrace** and **-rbrace** flags to override this precedence.

-public
Allows other users access to a sequence. The **-public** flag does not make protected messages available, only the sequence itself. This option is the default if the folder is not write-protected from other users.

-rbrace
Groups **-and**, **-or**, and **-not** operations. Operations between the **-lbrace** and **-rbrace** flags are evaluated as one operation. You can nest the **-lbrace** and **-rbrace** flags.

-search *"Pattern"*
Selects messages that contain the character string specified by the *"Pattern"* variable anywhere in the message.

-sequence *Name*
Stores the messages selected by the **pick** command in the sequence specified by the *Name* variable.

-to *"Pattern"*
Selects messages that contain the character string specified by the *"Pattern"* variable in the `To:` field.

-zero
Clears the specified sequence before placing the selected messages into the sequence. This flag is the default.

—*Component "Pattern"*
Selects messages that contain the character string specified by the *"Pattern"* variable in the heading field specified by the *Component* variable; for example, `pick –reply-to amy`.

Profile Entries

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

`Current-Folder:`	Sets the default current folder.
`Path:`	Specifies the user's MH directory.

Examples

1. To get a list of message numbers in the current folder that are from user `jones`, enter:

   ```
   pick  -from jones
   ```

   The system responds with a message similar to the following:

   ```
   12
   15
   19
   ```

2. To see a list of message numbers in the `schedule` folder received within the last 30 days, enter:

   ```
   pick  +schedule  -after -30
   ```

   The system responds with a message similar to the following:

   ```
   5
   8
   21
   30
   ```

Files

$HOME/.mh_profile	Contains the user's MH profile.
/usr/bin/pick	Contains the **pick** command.

Related Information

The **mark** command.

The **.mh_alias** file format, **.mh_profile** file format.

Mail Overview in *AIX 5L Version 5.2 System User's Guide: Communications and Networks*.

---

ping Command

Purpose

Sends an echo request to a network host.

Syntax

ping [**-d**] [**-D**] [**-n**] [**-q**] [**-r**] [**-v**] [**-R**] [**-a** *addr_family*] [**-c** *Count*] [**-w** *timeout*] [**-f** | **-i** *Wait*] [**-l** *Preload*] [**-p** *Pattern*] [**-s** *PacketSize*] [**-S** *hostname/IPv6 addr*] [**-L**] [**- I** *a.b.c.d.*] [**-o** *interface*] [**-T** *ttl*] *Host* [*PacketSize* [*Count*]]

Description

The **/usr/sbin/ping** command sends an Internet Control Message Protocol (ICMP) ECHO_REQUEST to obtain an ICMP ECHO_RESPONSE from a host or gateway. The **ping** command is useful for:

- Determining the status of the network and various foreign hosts.

- Tracking and isolating hardware and software problems.
- Testing, measuring, and managing networks.

If the host is operational and on the network, it responds to the echo. Each echo request contains an Internet Protocol (IP) and ICMP header, followed by a **timeval** structure, and enough bytes to fill out the packet. The default is to continuously send echo requests until an Interrupt is received (Ctrl-C).

The **ping** command sends one datagram per second and prints one line of output for every response received. The **ping** command calculates round-trip times and packet loss statistics, and displays a brief summary on completion. The **ping** command completes when the program times out or on receipt of a **SIGINT** signal. The *Host* parameter is either a valid host name or Internet address.

By default, the **ping** command will continue to send echo requests to the display until an Interrupt is received (Ctrl-C). The Interrupt key can be changed by using the **stty** command.

Because of the load that continuous echo requests can place on the system, repeated requests should be used primarily for problem isolation.

Flags

-c *Count* Specifies the number of echo requests, as indicated by the *Count* variable, to be sent (and received).

-w *timeout* This option works only with the -c option. It causes ping to wait for a maximum of 'timeout' seconds for a reply (after sending the last packet).

-d Starts socket-level debugging.

-D This option causes a hex dump to standard output of ICMP ECHO_REPLY packets.

-f Specifies flood-ping option. The **-f** flag ″floods″ or outputs packets as fast as they come back or one hundred times per second, whichever is more. For every ECHO_REQUEST sent, a . (period) is printed, while for every ECHO_REPLY received, a backspace is printed. This provides a rapid display of how many packets are being dropped. Only the root user may use this option.

> **Note:** This can be very hard on a network and should be used with caution. Flood pinging is only permitted by the root user. The **-f** flag is incompatible with the **-i** *Wait* flag.

-I *a.b.c.d* Specifies that the interface specified by *a.b.c.d* is to be used for outgoing IPv4 multicasts. The **-I** flag is an uppercase i.

-o *interface* Specifies that *interface* is to be used for outgoing IPv6 multicasts. The interface is specified in the form 'en0', 'tr0' etc.

-i *Wait* Waits the number of seconds specified by the *Wait* variable between the sending of each packet. The default is to wait for one second between each packet. This option is incompatible with the **-f** flag.

-L Disables local loopback for multicast pings.

-l *Preload* Sends the number of packets specified by the *Preload* variable as fast as possible before falling into normal mode of behavior (one per second). The **-l** flag is a lowercase l.

-n Specifies numeric output only. No attempt is made to look up symbolic names for host addresses.

-p *Pattern* Specifies up to 16 'pad' bytes to fill out the packet you send. This is useful for diagnosing data-dependent problems in a network. For example, **-p ff** fills the packet with all 1's.

-q Specifies quiet output. Nothing is displayed except the summary lines at startup time and when finished.

-r Bypasses the routing tables and sends directly to a host on an attached network. If the *Host* is not on a directly connected network, the **ping** command generates an error message. This option can be used to ping a local host through an interface that no longer has a route through it.

-R Specifies record route option. The **-R** flag includes the RECORD_ROUTE option in the ECHO_REQUEST packet and displays the route buffer on returned packets.

> **Note:** The IP header is only large enough for nine such routes. Also, many hosts and gateways ignore this option.

-a *addr_family* Maps the destination address of the ICMP packets to IPv6 format if addr_family is equal to ″inet6″.

-s *PacketSize* Specifies the number of data bytes to be sent. The default is 56, which translates into 64 ICMP data bytes when combined with the 8 bytes of ICMP header data.

-S *hostname/IPv6 addr* This is an IPv6 option. It allows you to set the source address in the ICMP packets to your scope preference (i.e., link local, site local, etc.).

-T *ttl* Specifies that the time-to-live for a multicast packet is *ttl* seconds.

-v Requests verbose output, which lists ICMP packets that are received in addition to echo responses.

Parameters

PacketSize Specifies the number of data bytes to be sent. The default is 56, which translates into 64 ICMP data bytes when combined with the 8 bytes of ICMP header data. This parameter is included for compatibility with previous versions of the **ping** command.

Count Specifies the number of echo requests to be sent (and received). This parameter is included for compatibility with previous versions of the **ping** command.

Examples

1. To check the network connection to host canopus and specify the number of echo requests to send, enter:

   ```
   ping  -c 5 canopus
   ```

   OR

   ```
   ping canopus  56  5
   ```

   Information similar to the following is displayed:

   ```
   PING canopus.austin.century.com: (128.116.1.5): 56 data bytes
   64 bytes from 128.116.1.5: icmp_seq=0 ttl=255 time=2 ms
   64 bytes from 128.116.1.5: icmp_seq=1 ttl=255 time=2 ms
   64 bytes from 128.116.1.5: icmp_seq=2 ttl=255 time=3 ms
   64 bytes from 128.116.1.5: icmp_seq=3 ttl=255 time=2 ms
   64 bytes from 128.116.1.5: icmp_seq=4 ttl=255 time=2 ms

   ----canopus.austin.century.com PING Statistics----
   5 packets transmitted, 5 packets received, 0% packet loss
   round-trip min/avg/max = 2/2/3 ms
   ```

2. To get information about host `lear` and start socket-level debugging, enter:

   ```
   ping  -d lear
   ```

   Information similar to the following is displayed:

   ```
   PING lear.austin.century.com: (128.114.4.18) 56 data bytes
   64 bytes from 128.114.4.18: icmp_seq=0 ttl=255 time=6 ms
   64 bytes from 128.114.4.18: icmp_seq=1 ttl=255 time=17 ms
   64 bytes from 128.114.4.18: icmp_seq=2 ttl=255 time=6 ms
   64 bytes from 128.114.4.18: icmp_seq=3 ttl=255 time=6 ms
   64 bytes from 128.114.4.18: icmp_seq=4 ttl=255 time=6 ms
   ```

```
^C
----lear.austin.century.com PING Statistics ----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 6/8/17 ms
```

Note: The output is repeated until an Interrupt (Ctrl-C) is received.

3. To obtain information about host `opus` and specify the number of data bytes to be sent, enter:

   ```
   ping  -s 2000 opus
   ```

   OR

   ```
   ping opus  2000
   ```

   Information similar to the following is displayed:

   ```
   PING opus.austin.century.com: (129.35.34.234): 2000 data bytes
   2008 bytes from 129.35.34.234: icmp_seq=0 ttl=255 time=20 ms
   2008 bytes from 129.35.34.234: icmp_seq=1 ttl=255 time=19 ms
   2008 bytes from 129.35.34.234: icmp_seq=2 ttl=255 time=20 ms
   2008 bytes from 129.35.34.234: icmp_seq=3 ttl=255 time=20 ms
   2008 bytes from 129.35.34.234: icmp_seq=4 ttl=255 time=20 ms
   2008 bytes from 129.35.34.234: icmp_seq=5 ttl=255 time=19 ms
   2008 bytes from 129.35.34.234: icmp_seq=6 ttl=255 time=19 ms
   ^C
   ----opus.austin.century.com PING Statistics----
   7 packets transmitted, 7 packets received, 0% packet loss
   round-trip min/avg/max = 19/19/20 ms
   ```

   Note: The output is repeated until an Interrupt (Ctrl-C) is received.

4. To invoke the flood-ping option to host `stlopnor`, enter:

   ```
   ping  -f stlopnor
   ```

   Information similar to the following is displayed:

   ```
   Ping stlopnor.austin.century.com: (129.35.34.234): 56 data bytes
   .^C
   ----stlopnor.austin.century.com PING Statistics ----
   1098 packets transmitted, 1097 packets received, 0% packet loss
   round-trip min/avg/max = 4/4/11
   ```

   Note: The flood-ping output continues until an Interrupt (Ctrl-C) is received.

5. To specify an interval of five seconds between packets sent to host `opus`, enter:

   ```
   ping  -i5 opus
   ```

   Information similar to the following is displayed:

   ```
   PING opus.austin.century.com: (129.35.34.234): 56 data bytes
   64 bytes from 129.35.34.234: icmp_seq=0 ttl=255 time=5 ms
   64 bytes from 129.35.34.234: icmp_seq=1 ttl=255 time=5 ms
   64 bytes from 129.35.34.234: icmp_seq=2 ttl=255 time=6 ms
   ^C
   ----opus.austin.century.com PING Statistics----
   3 packets transmitted, 3 packets received, 0% packet loss
   round-trip min/avg/max = 5/5/6 ms
   ```

   Note: The output is repeated until an Interrupt (Ctrl-C) is received.

6. To send the number of packets specified by the *Preload* variable as fast as possible before falling into normal mode of behavior to host `opus`, enter:

```
ping  -l 10 opus
```

Information similar to the following is displayed:

```
PING opus.austin.century.com: (129.35.34.234): 56 data bytes
64 bytes from 129.35.34.234: icmp_seq=0 ttl=255 time=9 ms
64 bytes from 129.35.34.234: icmp_seq=1 ttl=255 time=11 ms
64 bytes from 129.35.34.234: icmp_seq=2 ttl=255 time=16 ms
64 bytes from 129.35.34.234: icmp_seq=3 ttl=255 time=22 ms
64 bytes from 129.35.34.234: icmp_seq=4 ttl=255 time=26 ms
64 bytes from 129.35.34.234: icmp_seq=5 ttl=255 time=27 ms
64 bytes from 129.35.34.234: icmp_seq=6 ttl=255 time=30 ms
64 bytes from 129.35.34.234: icmp_seq=7 ttl=255 time=31 ms
64 bytes from 129.35.34.234: icmp_seq=8 ttl=255 time=33 ms
64 bytes from 129.35.34.234: icmp_seq=9 ttl=255 time=35 ms
64 bytes from 129.35.34.234: icmp_seq=10 ttl=255 time=36 ms
64 bytes from 129.35.34.234: icmp_seq=11 ttl=255 time=6 ms
64 bytes from 129.35.34.234: icmp_seq=12 ttl=255 time=6 ms
64 bytes from 129.35.34.234: icmp_seq=13 ttl=255 time=6 ms
64 bytes from 129.35.34.234: icmp_seq=14 ttl=255 time=7 ms
64 bytes from 129.35.34.234: icmp_seq=15 ttl=255 time=6 ms
^C
----opus.austin.century.com PING Statistics----
16 packets transmitted, 16 packets received, 0% packet loss
round-trip min/avg/max = 6/19/36 ms
```

Note: The output is repeated until an Interrupt (Ctrl-C) is received.

7. To diagnose data-dependent problems in a network, enter:

```
ping  -p ff opus
```

This command sends packets with a pad-pattern of all 1's to host `opus`. Information similar to the following is displayed:

```
PATTERN: 0xff
PING opus.austin.century.com: (129.35.34.234): 56 data bytes
64 bytes from 129.35.34.234: icmp_seq=0 ttl=255 time=5 ms
64 bytes from 129.35.34.234: icmp_seq=1 ttl=255 time=5 ms
64 bytes from 129.35.34.234: icmp_seq=2 ttl=255 time=5 ms
64 bytes from 129.35.34.234: icmp_seq=3 ttl=255 time=6 ms
64 bytes from 129.35.34.234: icmp_seq=4 ttl=255 time=5 ms
^C
----opus.austin.century.com PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 5/5/6 ms
```

Note: The output is repeated until an Interrupt (Ctrl-C) is received.

8. To specify quiet output, enter:

```
ping  -q bach
```

Only summary information similar to the following is displayed:

```
PING bach.austin.century.com: (129.35.34.234): 56 data bytes
^C
----bach.austin.century.com PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 5/5/8 ms
```

Note: Although not displayed, the output of packets continues until an Interrupt (Ctrl-C) is received.

Related Information

The **ifconfig** command, **netstat** command.

Network Overview in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

---

pioattred Command

Purpose

Provides a way to format and edit attributes in a virtual printer.

Syntax

pioattred -q *PrintQueueName* **-d** *QueueDeviceName* [**-o** *Action*] [**-a** *Attribute*]

Description

The **pioattred** command provides a way to format virtual printer attributes and to edit the attributes. Specifically, attributes in the printer definition file can be formatted and/or edited according to the action specified with the **-o** flag. Formatted attributes are written to standard output **stdout**. Attributes are edited with the editor specified in the **VISUAL** environment variable. The virtual printer definition file is assumed to be in the **/var/spool/lpd/pio/@local/custom/*** directory.

Flags

-a *Attribute*
Specifies the name of the attribute in the virtual printer definition file to format or edit. This flag may be specified many times.

-d *QueueDeviceName*
Specifies the *QueueDeviceName* spooler of the virtual printer definition to format or edit.

-o *Action*
Specifies the action that the **pioattred** command should take on the virtual printer definition. If this flag is omitted, the **pioattred** command assumes a value of 0 (zero).

- **0** Format the attributes specified. The result goes to **stdout**.
- **1** Format and edit the attribute(s) specified; use the editor specified in the **VISUAL** environment variable. If no editor is specified in the **VISUAL** environment variable, use the vi editor. If an error is made in editing the attributes, save the erroneous attributes in a temporary file, and return a return code indicating an error.

The following values are used in the event that an error return code was returned after editing the attributes.

- **2** Edit the attributes again. The virtual printer definition will be the state it was left in when the error occurred.
- **3** Ignore the error and save the edited attributes in the virtual printer definition.
- **4** Clean up and leave things in the state they were before the **pioattred** command was started.

-q *PrintQueueName*
Specifies the *PrintQueueName* spooler of the virtual printer definition to format or edit.

Examples

1. To format the **ci** and **sh** attributes in the queue: quedev virtual printer definition, enter:

```
pioattred -q queue -d quedev -o 0 -a ci -a sh
```

OR

```
pioattred -q queue -d quedev -a ci -a sh
```

2. To format all attributes in the queue: quedev virtual printer definition, enter:

```
pioattred -q queue -d quedev -o 0
```

OR

```
pioattred -q queue -d quedev
```

3. To edit the **st** attribute in the queue: quedev virtual printer definition, enter:

```
pioattred -q queue -d quedev -o 1 -a st
```

Files

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

Related Information

The **lsvirprt** command.

Virtual Printer Definitions and Attribute Values in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

---

piobe Command

Purpose

Print job manager for the printer backend.

Syntax

/usr/lpd/piobe [**-a** *PreviewLevel*] [**-A** *DiagnosticLevel*] [**-d** *InputDataStream*] [**-f** *FilterName*] [*FormatterFlags*] [*File* ...]

Description

The **piobe** command is a spooler backend program called by the **qdaemon** program to process a print job. The **piobe** command serves as the print job manager.

Based on the argument of the **-d** flag (or its default value in the database), which specifies the data stream type of the print files, the **piobe** command retrieves a pipeline from the database and passes it to a shell. The pipeline contains a string of filters to convert the print files as necessary and send them to a printer. If requested, the **piobe** command also retrieves and runs pipelines from the database to generate header and trailer pages.

The *FormatterFlags* argument (flags other than the flags listed in this article) is assumed to be referenced by the filter commands in the pipelines. If a flag is specified but not referenced by the pipelines, an error message is issued and the print job ended.

> **Note:** The **piobe** command should not be typed directly on the command line. This command is invoked by the **qdaemon** process and is dependent on the various services provided by the **qdaemon** process.

Flags

-a *PreviewOption*

Provides a way to preview parameter values that would be used for a print job without actually printing any files. Values that can be specified for the *PreviewOption* variable are:

0 Specifies normal print processing

1 Returns a list of flag values and the pipeline of filters that would be used to convert the input data type to the data type expected by the printer, but does not actually invoke the pipeline of filters or send the file to the printer.

The list of flag values returned are the default command line flag values from the configuration database. These values are overridden by any flag arguments specified on the command line. Please note that:

- Only flags that are valid for the *InputDataType* variable specified (or defaulted) for the **-d** flag are shown.
- Flag values related only to the spooling of your print job, instead of the actual printing, are not shown. The default values for the spooling flags are included with the descriptions of the flags for the **qprt** command.
- The flag values may not have been checked to verify that they are valid.

The pipeline of filters shows the filter commands (and the flag values passed to the filter commands) that would process the data from your print file before it is sent to the printer. You can review the description for each of the filter commands to determine the type of filtering that would be performed.

-A *Value*

Specifies the level of diagnostic output. Diagnostic output is useful for diagnosing errors encountered by a pipeline of filters that is processing a print file, a header page, or a trailer page. Diagnostic output is mailed to the user who submitted the print job. The *Value* variable can be one of the following:

0 Discards any standard error output that is produced by the header, trailer, or print file pipelines.

1 If any standard error output is produced, returns the standard error output and the pipeline that produced it and ends the print job.

2 Returns the flag values, standard error output (if any), and completes pipelines, regardless of whether an error is detected. If an error is detected, the print job is ended.

3 Similar to a value of **2**, except that the file is not printed.

A value of **1** is recommended. A value of **0** is used if a filter in a pipeline produces output to standard error, even if no error is encountered, such as for status information. A value of **2** or **3** is used for diagnosing a problem even if the problem does not cause any output to standard error.

-d *InputDataType*

Specifies the type of data that is in the file to be printed. This flag is a one-character identifier. Based on the data type for the print file and the data type expected by the printer, the print files are passed through filters (if necessary) before being sent to the printer. Examples of data type identifiers are:

a IBM extended ASCII

p Pass-through (sent to the printer unmodified)

s PostScript

c PCL

d Diablo 630

k Kanji.

If the printer you select does not support the *InputDataType* variable and filters are not available to convert the data type of your print file to a data type supported by the printer, the print job will be ended with an error message.

-f *FilterType*

Specifies a type of filter through which your print file is passed before being sent to the printer. This flag is a one-character identifier. The identifiers are similar to the filter flags available with the **lpr** command. The available filter identifier is **p** which invokes the **pr** filter.

Files

/var/spool/lpd/pio/@local/ddi/* Contains the digested database files.

Related Information

The **lpr** command, **pr** command, **qdaemon** command, **qprt** command.

The **qconfig** file.

Printer Backend Overview for Programming in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Understanding the Interaction between qdaemon and the Backend in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Understanding Backend Routines in libqb in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Printer Addition Management Subsystem: Programming Overview in *AIX 5L Version 5.2 Kernel Extensions and Device Support Programming Concepts*.

Configuring a Printer without Adding a Queue in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Adding a Printer Using the Printer Colon File in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

---

pioburst Command

Purpose

Generates burst pages (header and trailer pages) for printer output.

Syntax

/usr/lpd/pio/etc/pioburst [**-H** *HostName*] *TextFile*

Description

The **pioburst** command retrieves prototype text for a burst page from the file specified by the *TextFile* variable, fills in the variable fields identified by **%** escape sequences in the prototype text, and writes the constructed text to standard output. It is invoked as a filter in a pipeline by the print job manager, the **piobe** command.

The **%** escape sequences, which are replaced by corresponding values, are:

%A Specifies the formatting flag values.
%D Specifies the user to whom the print output is to be delivered.
%H Specifies the name of the host machine printing the job.
%P Specifies the time the print job was printed.
%Q Specifies the time the print job was queued.
%S Specifies the user who submitted the print job.
%T Specifies the title of the print job.
%% Specifies the % (percent sign).

Labels (20 characters long) for each of the variable fields can be specified by using the same escape sequence as for the variable field, except using lowercase letters. For example, to generate a label for the variable field specifying the print job was queued (**%Q**), use **%q**. The **%e** variable represents the label `END OF OUTPUT FOR:`.

The **pioburst** command requires the following environment variables to be initialized:

PIOTITLE Title of the print job (for **%T**)
PIOQDATE Time the print job was queued (for **%Q**)
PIOFROM User who submitted the print job (for **%S**)
PIOTO User to whom the print output is to be delivered (for **%D**)
PIOFLAGS Flag values (for **%A**).

Flags

-H *HostName* Specifies that the host name designated by the *HostName* variable override the default host name (the name of the host machine printing the job).

Example

To generate a header page and send it to standard output, enter:

```
pioburst /usr/lpd/pio/burst/H.ascii
```

Files

/usr/lpd/pio/etc/pioburst Contains the **pioburst** command.

Related Information

The **digest** command, **piobe** command, **piodigest** command, **pioformat** command, **piofquote** command, **pioout** command, **piopredef** command.

The Printer Overview for System Management in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Printer Colon File Conventions in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Printer Colon File Escape Sequences in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Printer Code Page Translation Tables in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Virtual Printer Definitions and Attributes in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

---

piocnvt Command

Purpose

Expands or contracts a predefined printer definition or a virtual printer definition.

Syntax

piocnvt [**-s** *State*] **-i** *SourceFile* [**-o** *TargetFile*]

Description

The **piocnvt** command takes either a predefined printer definition or a virtual printer definition and expands or contracts the file. An expanded printer definition file contains all the attributes associated with that printer definition. A contracted printer definition contains only the printer specific attributes for that printer definition.

Printer definition files are arranged in a hierarchical parent-child relationship. For example the predefined printer definition `4201-3.asc` has the parent `master`. An expanded printer definition for `4201-3.asc` would contain all the attributes from `4201-3.asc` as well as those from `master`. A contracted printer definition for `4201-3.asc` would contain only the attributes not found in `master`. The **piocnvt** command simply provides a way to move back and forth between the expanded and contracted states of a printer definition file.

Flags

-i *SourceFile*
: Specifies the complete path and name of the input file.

-o *TargetFile*
: Specifies the complete path and name of the output file. If the **-o** flag is omitted, the *SourceFile* will be used for output.

-s *State*
: Specifies whether the state of the *TargetFile* parameter should be expanded or contracted. If the **-s** flag is omitted, the **piocnvt** command attempts to determine the state by examining the **zD** attribute in the *SourceFile*. If a determination cannot be made the *TargetFile* parameter will be left in an expanded state.

  + Indicates that the state of the *TargetFile* parameter should be expanded.

  ! Indicates that the state of the *TargetFile* parameter should be contracted.

Examples

1. To expand the virtual printer definition `lp0:lp0` into the file `new:lp0`; enter:

   ```
   piocnvt -s+ -i lp0:lp0 -o new:lp0
   ```

2. To contract the virtual printer definition `lp0:lp0` in place; enter:

   ```
   piocnvt -s! -i lp0:lp0
   ```

Files

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

Related Information

The **chvirprt** command, **lsvirprt** command, **mkvirprt** command.

Printer Overview for System Management in the *AIX 5L Version 5.2 Guide to Printers and Printing*

Spooler Overview for System Management in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Printer Specific Information in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Printer Support in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Virtual Printer Definitions and Attributes in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

---

piodigest Command

Purpose

Digests attribute values for a virtual printer definition into a memory image and stores the memory image in a file.

Syntax

/usr/lpd/pio/etc/piodigest [**-s** *DataStreamType*] [**-n** *DeviceName*] [**-p** *DirectoryPath*] [**-q** *PrintQueueName*] [**-t** *PrinterType*] [**-d** *QueueDeviceName*] { *ColonFileName* | **-** }

Description

The **piodigest** command retrieves virtual printer attribute values from a colon file, builds a memory image of the attribute values and their lookup tables, and writes the constructed memory image to a file. The memory image in the file is then available for access by a print formatter and by the **piobe** command (the print job manager). The command also creates Object Data Manager (ODM) stanzas for the specified queue and queue devices. The ODM stanzas are used in System Management Interface Tool (SMIT) dialogs. If an attribute called **zV** is specified and the attribute contains a value of **+**, the **piodigest** command performs syntax, reference, and limits validation on all attributes specified in the colon file.

The **piodidgest** command should be invoked whenever a customized version of a virtual printer definition is initially generated or is later modified. Each invocation of the **piodigest** command digests the attribute values for one virtual printer definition.

The *ColonFileName* parameter is the name of the input file in colon format. A colon file contains the attribute values for one virtual printer. A value of **-** (dash) for the *ColonFileName* parameter indicates that the colon file should be read from standard input.

The name of the output file that is generated will be of the form:

```
PrinterType.DataStreamType.DeviceName.PrintQueueName:QueueDeviceName
```

Flags

-d *QueueDeviceName*
: Specifies the name of the virtual printer (queue device).

  If this flag is not specified, the virtual printer name specified by the **mv** attribute from the input colon file is assumed.

-n *DeviceName*
: Specifies the name of the printer device, such as `lp0` for line printer 0, or `lp1` for line printer 1.

  If this flag is not specified, the device name specified by the **mn** attribute from the input colon file is assumed.

-p *DirectoryPath* Specifies the path name of the directory where the output file is to be generated.

If this flag is not specified, the **/var/spool/lpd/pio/@local/ddi** directory is assumed.

-q *PrintQueueName* Specifies the name of the print queue to which the virtual printer is assigned.

If this flag is not specified, the print queue name specified by the **mq** attribute from the input colon file is assumed.

-s *DataStreamType* Specifies the printer data stream type. Example data stream types are **asc** (IBM extended ASCII), **ps** (PostScript), **pcl** (HP PCL), and **630** (Diablo 630).

If this flag is not specified, the data stream type specified by the **md** attribute from the input colon file is assumed.

-t *PrinterType* Specifies the printer type. Examples are `4201-3` and `ti2115`.

If this flag is not specified, the printer type specified by the **mt** attribute from the input colon file is assumed.

Example

To generate a digested virtual printer definition, enter:

```
piodigest -d mypro -n lp0 -q proq -s asc -t 4201-3
```

The attribute values for the virtual printer assigned to the `mypro` queue device on the `proq` print queue are digested and stored in the file named **4201-3.asc.lp0.proq:mypro** in the **/var/spool/lpd/pio/@local/ddi** directory.

Files

/var/spool/lpd/pio/@local/ddi/* Contains the digested, virtual printer definitions.
/usr/lpd/pio/etc/piodigest Contains the **piodigest** command.

Related Information

The **mkvirprt** command, **piobe** command, **pioburst** command, **pioformat** command, **piofquote** command, **pioout** command, **piopredef** command.

The Printer Overview for System Management and Virtual Printer Definitions and Attributes, Printer Code Page Translation Tables, Printer Colon File Conventions, Printer Colon File Escape Sequences, Printer Backend Overview for Programming, Understanding the Interaction between qdaemon and the Backend, Understanding Backend Routines in libqb in *AIX 5L Version 5.2 Guide to Printers and Printing*.

---

piodmgr Command

Purpose

Compacts the Object Data Manager (ODM) database in the **/var/spool/lpd/pio/@local/smit** directory.

Syntax

piodmgr { **-c** | **-h** }

Description

The **piodmgr** command extracts existing printer definitions from the ODM database in the **/var/spool/lpd/pio/@local/smit** directory, recreates the ODM database, compacts the database, and reloads the compacted database.

The **-c** and **-h** flags are mutually exclusive. The **-h** flag only compacts the database when the host name has been changed. The **-c** flag always compacts the database.

Note: Root user authority is needed to run this command.

Flags

-c Extracts existing printer definitions from the ODM database, recreates the database, compacts the information, and replaces the database.

-h Performs exactly like the **-c** flag, but the **-h** flag compacts the information only if the host name has been changed. If the host name has been changed, the **-h** flag extracts the new name and updates the host name information in the database. If the host name has not been changed, the **-h** flag does not compact the information. This flag is an optional compactor rather than an automatic compactor as with the **-c** flag.

Examples

1. To compact and update the ODM printer definition database, enter:

   ```
   piodmgr -c
   ```

2. To perform compaction of the information depending on whether the host name has been changed or not, enter:

   ```
   piodmgr -h
   ```

Files

/usr/lib/lpd/pio/etc/piodmgr	Contains the **piodmgr** command.
/var/spool/lpd/pio/@local/smit/*	Contains predefined printer definitions used by the command.

Related Information

The **piobe** command.

Printer Overview for System Management, Spooler Overview for System Management, and, Printer Backend Overview for Programming in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Object Data Manager (ODM) Overview for Programmers in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

---

piofontin Command

Purpose

Copies fonts from a multilingual font diskette.

Syntax

piofontin -t *PrinterType* **-c** *Codepage* [**-d** *Device*]

Description

The **piofontin** command copies font files from a multilingual font diskette to a directory one level beneath the **/usr/lib/lpd/pio/fonts** label. The directory to which the font files are copied has the name specified by the *PrinterType* parameter. The font files are named according to the naming convention for files. Names are of the form:

```
codepage.typeface.pitch*10.quality
```

Only the root user can use the **piofontin** command.

Flags

-c *Codepage*	Specifies the code page for the fonts. For Greek fonts the value is 851, and for Turkish fonts the value is 853.
-d *Device*	Specifies the diskette-drive device name. This defaults to the **-d/dev/fd0** label, the standard 3.5-inch diskette drive.
-t *PrinterType*	Specifies the type of printer for the fonts. Supported printer types are 4201-3, 4202-3, 4207-2, 4208-2, 2380, 2381, 2390, and 2391.

Example

To read a diskette containing 4201-3 fonts in code page 851 from diskette drive `/dev/fd1`**;** enter**:**

```
piofontin 4201-3 851 /dev/fd1
```

The font files are copied to the **/usr/lib/lpd/pio/fonts/4201-3** directory.

File

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

Related Information

Printer Overview for System Management, Printer Specific Information, Printer Support, Virtual Printer Definitions and Attributes, Printer Code Page Translation Tables in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

---

pioformat Command

Purpose

Drives a printer formatter.

Syntax

/usr/lpd/pio/etc/pioformat -@ *DataBaseFile* [**-!** *FormatterName*] [**-# +** *PassThroughOption*]

Description

The **pioformat** command initiates the printer formatter driver. The formatter driver establishes access to the database values, loads and links a printer formatter, and then drives the formatter by calling its **setup** function, **initialize** function, **lineout** function, **passthru** function, and **restore** function as appropriate. The formatter driver also provides the **piogetvals** subroutine, **piogetopt** subroutine, **piocmdout** subroutine, **piogetstr** subroutine, **piomsgout** subroutine, and **pioexit** subroutine used by the formatter.

The flags listed below are processed by the formatter driver and are not passed on to the formatter. However, all flags NOT listed below are assumed to be formatting flags and are passed on to the formatter.

Flags

-@ *DataBaseFile*

Specifies either of the following:

- The full path name of the (digested) database file to be accessed
- The print queue and queue device names, separated by a colon

If the argument string begins with a **/** (slash) character, it is assumed to be a full path name.

The combination of the queue name and the queue device name results in a unique string that is a part of the database file name and is used to search for the database file name in the **/var/spool/lpd/pio/@local/ddi** directory. This short form alternative is provided as a convenience when the formatter driver and formatter are run as standalone devices, instead of by the spooler.

-! *FormatterName*

Specifies the full path name of the formatter to be loaded, linked, and driven.

If the **-!** flag is not specified, the default formatter name defined by the **mf** attribute name in the database is used. A default formatter name is provided as a convenience when the formatter driver and formatter are run as standalone devices, instead of by the spooler.

-# + *PassThroughOption*

Specifies that the print file should be passed through unmodified. If the **-# +** flag is not specified, the print file will be formatted.

The parameter that is passed to the formatter's **setup** routine contains a value of 1 instead of 0, indicating that the file should be passed through instead of being formatted.

Examples

1. To format the `myfile` file according to the database file (virtual printer description) for the queue device named `std` associated with the print queue named `pro`, overriding the page width to 132 characters, and using the **pioformat** command and a formatter as a standalone filter, enter:

   ```
   cat myfile | pioformat  -@ pro:std -w 132 >/dev/lp0
   ```

2. To use the **pioformat** command and a formatter in a pipeline running under the spooler, enter:

   ```
   %Ide/pioformat  -@ %Idd/%Imm  -! %Idf/piof420x %Fbb %Fee ...
   ```

   For this example, assume that:

   - The printer is a 4207 Model 2 Proprinter.
   - The print queue name is `pro`.
   - There is only one queue device (virtual printer) defined for the print queue and its name is `std` and its output data stream type is `asc` (extended ASCII).
   - The printer device name is `/dev/lp0`.
   - The print job submitter specified the flag and argument `-i 5`.

   Before the print job manager (the **piobe** command) passes the pipeline to a shell to format the file, it resolves the pipeline's embedded references to attribute values. Based on the assumptions listed above for this example, the attribute references would be resolved as:

   ```
   %Ide -> /usr/lpd/pio/etc
   ```

   Directory where the **pioformat** command resides

   ```
   %Idd -> /var/spool/lpd/pio/@local/ddi
   ```

   Directory for database files

   ```
   %Imm -> 4207-2.asc.lp0.pro:std
   ```

	Database file name
`%Idf -> /usr/lpd/pio/fmtrs`	Directory for formatters
`%Fbb ->`	Null string, since submitter did not specify the **-b** flag
`%Fee -> -i 5`	Submitter specified this flag and argument.

The resulting pipeline shown below would be passed to a shell to format the file (shown on multiple lines for readability):

```
/usr/lpd/pio/etc/pioformat        # initiate the formatter driver
-@/usr/lpd/pio/ddi/4207-2.asc.lp0.pro:std
                                  # (digested) database file
-!/usr/lpd/pio/fmtrs/piof420x     # loadable formatter
-i5                               # formatting option
                                  # (indent 5 characters)
```

Files

/usr/lpd/pio/etc/pioformat	Contains the formatter driver.
/usr/lpd/pio/fmtrs/*	Contains the formatters.
/var/spool/lpd/pio/@local/ddi/*	Contains the digested database files.

Related Information

The **piobe** command, **pioburst** command, **piodigest** command, **piofquote** command, **pioout** command, **piopredef** command.

The **piocmdout** subroutine **piogetvals** subroutine, **piogetopt** subroutine, **piogetstr** subroutine, **piomsgout** subroutine, **pioexit** subroutine.

Printer Overview for System Management in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Virtual Printer Definitions and Attributes in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Printer Addition Management Subsystem: Programming Overview in *AIX 5L Version 5.2 Kernel Extensions and Device Support Programming Concepts*.

Printer Code Page Translation Tables in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Printer Colon File Conventions in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Printer Colon File Escape Sequences in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Printer Specific Information in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Printer Support in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Printer Backend Overview for Programming in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Adding a Printer Using the Printer Colon File in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

piofquote Command

Purpose
Converts certain control characters destined for PostScript printers.

Syntax
/usr/lpd/pio/etc/piofquote

Description
The **piofquote** command is a filter that converts certain control characters destined for PostScript printers that can emulate other printers. The command reads data from standard input, checks for control characters, and modifies them as needed. It then writes the data to standard output.

If a least 1 byte of data appears on standard input, the **piofquote** command writes a hex 04 control character to standard output before the first input data byte is written to standard output. The command also writes a hex 04 to standard output when end-of-file is recognized on standard input.

If a hex 01, 03, 04, 05, 11, 13, 14, or 1c control character is found in the input data read from standard input, the hex 40 bit in the control character is turned on and a hex 01 character is prefixed to the control character before it is written to standard output.

Files

standard input	Input data stream to be processed.
standard output	Output data stream containing converted control characters.

Related Information
The **piobe** command, **pioburst** command, **piodigest** command, **pioformat** command, **pioout** command, **piopredef** command.

Printer Overview for System Management in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

piolsvp Command

Purpose
Lists virtual printers on a system.

Syntax
piolsvp { **-q** | **-v** | **-Q** | **-p** | **-A** } [**-n***AttachmentField*]

piolsvp -P *Queue* [**:** *QueueDevice*] **-n***AttachmentField*

piolsvp -P *Queue* **-d**

piolsvp -N *AttachmentType* **-n***AttachmentField*

Description
The **piolsvp** command lists the virtual printers and attachment types on the system. The **piolsvp** command displays either the queues or the queues plus the queue-device pairs for virtual printers.

The order of the list of queues and queue-device pairs is the same as the order used by the **/etc/qconfig** file.

Flags

-A Displays all attachment types and descriptions for the attachment types. The **.attach** and **.config** files in the **/usr/lib/lpd/pio/etc** directory define all attachment types.

-d Displays the queue devices associated with a given queue.

-n*AttachmentField* Specifies a field name for an attachment. The field name is typically a SMIT selector name. Possible values for the *AttachmentField* variable are:

submit_job

add_queue

add_printer

remove_queue

printer_conn

change_queue

change_filters

When the **-n** and **-A** flags are specified, only the attachment types that have a value for the specified attachment field in their attachment files are displayed. Attachment definitions are kept in the files with the *AttachmentType***.attach** naming convention. The **.attach** files reside in the **/usr/lib/lpd/pio/etc** directory.

When the **-n** flag is specified with either the **-q** or **-v** flags, only queues and queue-device pairs that belong to defined attachment types are displayed. A defined attachment type has an assigned field value in the definition files.

When the **-n** flag is specified with the **-P** flag, the SMIT selector name is displayed. The **-n** and **-P** flag combination also displays the queue device name and attachment type.

When the **-n** flag is specified with the **-N** flag, the SMIT selector name is displayed for the specified attachment field and attachment type.

-N Specifies an attachment type. The SMIT selector name associated with a given attachment field is displayed.

-p Displays all the queue and queue-device pairs on the system and provides a description of each queue and queue-device pair. Only the queue name for the first queue-queue is displayed if there are queues with multiple queue devices.

-P Specifies the queue name or queue device name for which information is displayed. The information consists of queue device name, attachment type, and SMIT selector value name.

-q Displays all queues on the system. The **-q** flag also displays the queue-device pairs for queues that have more than one device.

-Q Displays all the queues on the system. The **-Q** flag does not list queue-device pairs. Use the **-q** flag to list queue-device pairs.

-v Displays all queue-device pairs for the queues that have virtual printers.

Examples

1. To display all the print queues on the system, enter:

```
piolsvp -q
```

The output of this command is:

```
e4019a                  4019 (IBM ASCII)
d3816                   IBM 3816 Page Printer
ena_asc                 4029 (IBM ASCII)
ena_gl                  4029 (Plotter Emulation)
ena_pcl                 4029 (HP LaserJet II Emulation)
ena_ps                  4029 (PostScript)
hplj2                   Hewlett-Packard LaserJet II
tstx                    4216-31 (Proprinter XL Emulation)
e4019ps                 4019 (PostScript)
4019lxxa                4029 (PostScript)
4019lxxa:lxx            4029 (PostScript)
40191xxa:rkmlxx         4019 (IBM ASCII)
4019lxxa:rkmlxxl        4019 (IBM ASCII)
```

2. To display all the virtual printers in the system, enter:

```
piolsvp -v
```

The output of this command is:

```
#QUEUE          DEVICE         DESCRIPTION
e4019a          e4019          4019 (IBM ASCII)
d3816           ena3816        IBM 3816 Page Printer
ena_asc         ena            4029 (IBM ASCII)
ena_gl          ena            4029 (Plotter Emulation)
ena_pcl         ena            4029 (HP LaserJet II Emulation)
ena_ps          ena            4029 (PostScript)
hplj2           lxx            Hewlett-Packard LaserJet II
tstx            lxx            4216-31 (Proprinter XL Emulation)
e4019ps         e4019          4019 (PostScript)
4019lxxa        lxx            4029 (PostScript)
4019lxxa        rkmlxx         4019 (IBM ASCII)
40191xxa        rkmlxx         4019 (IBM ASCII)
```

3. To list all the queues on the system, enter:

```
piolsvp -Q
```

The output of this command is:

```
e4019a          4019 (IBM ASCII)
d3816           IBM 3816 Page Printer
ena_asc         4029 (IBM ASCII)
ena_gl          4029 (Plotter Emulation)
ena_pcl         4029 (HP LaserJet II Emulation)
ena_ps          4019 (PostScript)
hplj2           Hewlett-Packard LaserJet II
tstx            4216-31 (Proprinter XL Emulation)
e4019ps         4019 (PostScript)
4019lxxa        4029 (PostScript)
```

4. To list all the attachment types that have a SMIT selector value specified for the `add_queue` SMIT selector, enter:

```
piolsvp -A -nadd_queue
```

The output from this command is:

```
#ATTACHMENT TYPE        DESCRIPTION
local                   Local Attached
remote                  Remote Attached
ascii                   ASCII Terminal Attached
other                   Generic Backend Attached
```

5. To list information for the `4019lxxa` queue, enter:

```
piolsvp -P4019lxxa -n add_queue
```

The output from this command is:

```
lxx     xsta    sm_xsta_addq_sel
```

6. To list the SMIT selector value for the remote attachment, enter:

   ```
   piolsvp -Axst -nadd_queue
   ```

   The output from this command is:

   ```
   sm_xsta_addq_sel
   ```

Files

/usr/lib/lpd/pio/etc/piolsvp	Contains the **piolsvp** command.
/etc/qconfig	Contains the configuration files.
/var/spool/lpd/pio/@local/custom/*	Contains the customized virtual printer attribute files.
/usr/lib/lpd/pio/etc/*.attach	Contains the attachment type files

Related Information

The **piobe** command, **qprt** command.

Printer Attachment Files in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Printer Overview for System Management in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Spooler Overview for System Management in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Printer Backend Overview for Programming in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

---

piomgpdev Command

Purpose

Manages printer pseudo-devices.

Syntax

piomgpdev -p *PseudoDevice* **-t** *AttachmentType* { **-A** | **-C** | **-R** | **-D** } [**-a** *Clause* ...]

Description

The **piomgpdev** command changes and removes pseudo-devices for printer attachments. The **piomgpdev** command stores information about the pseudo-devices in files in the **/var/spool/lpd/pio/@local/dev** directory. The file contains stanzas in the following form:

```
key_word = value
```

The information stored in these files pertains to connection characteristics for a given attachment and a printer.

Flags

-a *Clause*	Specifies a clause to be added or changed in the file for a pseudo-device. The clause is in the following form: `key_word = value` If the **-D** flag is specified, the clause can contain only the keyword.
-A	Adds a pseudo-device.
-C	Changes a pseudo-device.
-D	Displays information for a specified clause of a pseudo-device definition.

-p *PseudoDevice* Specifies the name of a pseudo-device for a printer attachment.
-R Removes a pseudo-device.

Files

/usr/lib/lpd/pio/etc/piomgpdev Contains the **piomgpdev** command.
/var/spool/lpd/pio/@local/dev/* Contains the printer pseudo-device files.

Related Information

The **piobe** command, **qprt** command.

Printer Overview for System Management in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Spooler Overview for System Management in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Printer Backend Overview for Programming in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

---

piomkapqd Command

Purpose

Builds a SMIT dialog to create print queues and printers.

Syntax

To Create a Print Queue for an Existing Printer

piomkapqd -A *AttachmentType* **-p** *Printer* **-d** *DeviceName* **-h** *Header* [**-e**]

To Create a Printer and a Print Queue

piomkapqd -A *AttachmentType* **-p** *Printer* **-v** *Device* **-s** *Subclass* **-r** *Adapter* **-h** *Header* [**-e**]

To Create a Printer Attached to a TTY or to Assign Printer Output to a File and Create a New Queue

piomkapqd -A *AttachmentType* **-p** *Printer* { **-T** *TTYName* | **-f** *FileName* } **-h** *Header* [**-e**]

To Use a User-Defined Attachment for a New Printer and Print Queue

piomkapqd -A *AttachmentType* **-p** *Printer* [**-d** *DeviceName*] **-c** *CmdExec* **-i** *DiscCmd* **-o** *ObjectID* **-h** *Header* [**-e**]

Description

The **piomkapqd** command creates a System Management Interface Tool (SMIT) dialog that allows the user to create new printers and print queues. The **piomkapqd** command also allows users to add their user-defined attachment types to a SMIT printer or queue definition dialog.

Flags

-A *AttachmentType* Specifies the type of attachment used to connect the printer to the data source. Common values for the *AttachmentType* variable are:

local Specifies a local attachment type.

ascii Specifies an ASCII attachment type.

file Specifies a file where the data is stored.

-c *CmdExec*	Specifies the value for the **cmd_to_execute** SMIT command. This flag is used when creating a user-defined attachment dialog. If this flag is not included, the **piomkpq** command is used as the default.
-d *DeviceName*	Specifies the name of the device, pseudo-device, or file where the output is directed, for example `lp0` or `tty1`.
-e	Specifies that an existing print queue is to be used for printer output. The **-e** prevents the **piomkapqd** command from creating a new queue.
-f *FileName*	Indicates the name of the file where output is stored.
-h *Header*	Specifies the title or header of the SMIT dialog that is being created.
-i *DiscCmd*	Specifies the value of the **cmd_to_discover** SMIT command. This flag is used when creating a user-defined attachment dialog. If this flag is not included, the **piomkapqd** command default value is used to create the dialog.
-o *ObjectID*	Specifies the SMIT object whose ID matches the value of the *ObjectID* variable.
-p *Printer*	Specifies the printer type as defined in the **/usr/lib/lpd/pio/predef** directory, for example `ibm4019`.
-r *ParentAdapter*	Specifies the parent adapter for the printer.
-s *Subclass*	Specifies the subclass type to which the printer belongs. The possible values for the *Subclass* variable are: • **parallel** • **rs232** • **rs422**
-T *TTYName*	Specifies the name of the TTY attached to the new printer or queue.
-v *Device*	Specifies the device type as defined in the ODM database. The **-v** flag retrieves printer definitions that are not stored in the **/usr/lib/lpd/pio/predef** directory.

Examples

1. To create a SMIT dialog that adds a print queue to an existing local printer, enter:
   ```
   piomkapqd -A local -p ibm4019 -d lp0 -h 'Add a New Queue'
   ```
2. To create a SMIT dialog that adds a new printer named `lp2` and new print queue attached locally, enter:
   ```
   piomkapqd -A local -p ibm4019 -v ibm4019 -s rs232 -r sa0 -h 'Add New Printer'
   ```
3. To create a SMIT dialog that adds a printer attached to a TTY and create a new queue for the printer, enter:
   ```
   piomkapqd -A tty -p ibm4039 -T tty12 -h 'Add TTY Printer'
   ```
4. To create a SMIT dialog that directs output to a file name `stuff` and to create a new queue, enter:
   ```
   piomkapqd -A file -p ibm4039 -f stuff -h 'Add Output File' -e
   ```
5. To create a SMIT dialog that adds a user-defined printer attachment type and creates a new queue, enter:
   ```
   piomkapqd -A hpJetDirect -p hplj-4 [-d lp0] -c /usr/sbin/mkjetd -i /usr/bin/lsjd -o JetDirect -h 'Add New Attachment Type'
   ```

File

/usr/lib/lpd/pio/etc/piomkapqd	Contains the **piomkapqd** command.

Related Information

The **piobe** command, **piomkpq** command .

Printer Overview for System Management, Queuing System Overview for System Management in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

System Management Interface Tool in the *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices*.

Printer Backend Overview for Programming in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

---

piomkpq Command

Purpose

Creates a print queue.

Syntax

To add a new printer

piomkpq -A *AttachmentType* **-p** *PrinterType* **-Q** *QueueName* **-D** *DataStream* **-v** *DeviceType* **-s** *Subclass* **-r** *ParentAdapter* **-w** *PortNumber* [
-a { *interface* | *ptop* | *autoconfig* | *speed* | *parity* | *bpc* | *stops* | *xon* | *dtr* | *tbc*=*DescValue* }] ...

To create a new print queue

piomkpq -A *AttachmentType* **-p** *PrinterType* { **-D** *DataStream* / **-q** *QueueName* } **-s** *Subclass* **-r** *ParentAdapter* **-w** *PortNumber* **-v** *DeviceType* [
-a { *interface* | *ptop* | *autoconfig* | *speed* | *parity* | *bpc* | *stops* | *xon* | *dtr* | *tbc*=*DescValue* }] ...

To create print queues for an existing printer

piomkpq -A *AttachmentType* **-p** *PrinterType* **-d** *DeviceName* { **-D** *DataStream* / **-q** *QueueName* }

To add an existing printer to an existing print queue

piomkpq -A *AttachmentType* **-p** *PrinterType* **-d** *DeviceName* **-D** *DataStream* **-q** *QueueName*

Description

The **piomkpq** command creates print queues and printers. This command is used by SMIT dialogs created with the **piomkapqd** command. The **piomkpq** command performs the following functions:

- Creates printer devices with various attachment types.
- Creates print queues.
- Creates queue devices.
- Creates virtual printers.
- Creates pseudo-devices.

Flags

-a Specifies a device attribute. This takes the form *Attribute=Value*, for example: `-a speed=9600`. The valid attributes are:

Interface

ptop

autoconfic

speed

parity

bpc

stops

xon

dtr

tbc

-A *AttachmentType* Specifies the type of attachment used to connect the printer to the data source. Common values for the *AttachmentType* variable are:

local Specifies a local attachment type.

ascii Specifies an ASCII attachment type.

file Specifies a file where the data is stored.

-d *DeviceName* Specifies the name of the device, pseudo-device, or file where the output is directed, for example `lp0` or `tty1`.

-D *DataStream* Specifies the datastream of a print queue to be created or an existing print queue.

-p *PrinterType* Specifies the printer type as defined in the **/usr/lib/lpd/pio/predef** directory, for example `ibm4019`.

-q *QueueName* Specifies a new queue name. The **-q** and **-Q** flags are exclusive.

-Q *QueueName* Specifies an existing queue name. The **-q** and **-Q** flags are exclusive.

-s *Subclass* Specifies the subclass type to which the printer belongs. The possible values for the *Subclass* variable are:

- **parallel**
- **rs232**
- **rs422**

-r *ParentAdapter* Specifies the parent adapter for the printer.

-w *PortNumber* Specifies the port number for the printer attachment.

-v *DeviceType* Specifies the device type as defined in the ODM database.

Examples

1. To create a local print queue named `castor` of datastream ASCII for an existing IBM 4019 printer named `lp0`, enter:

   ```
   piomkpq -A local -p ibm4019 -d lp0 -D asc -q castor
   ```

2. To add an existing local printer to an existing local print queue called `pyrite` for the datastream PostScript, enter:

   ```
   piomkpq -A local -p ibm4019 -d lp0 -Q pyrite -D ps
   ```

3. To create local print queue called `baker` for a new printer, enter:

   ```
   piomkpq -A local -p ibm4019 -D asc -Q baker -s parallel -r ppa0
   -w p -v ibm4019 [-a ptop=120]
   ```

4. To create the **clues** file print queue, enter:

   ```
   piomkpq -A file -p ibm4019 -d clues -D asc -q baker
   ```

Files

/usr/lib/lpd/pio/etc/piomkpq	Contains the **piomkpq** command.
/usr/lib/lpd/pio/etc/piomgpdev	Creates a pseudo-device.
/usr/sbin/mkdev	Creates a device.
/usr/bin/mkque	Creates a queue.
/usr/bin/mkquedv	Creates a queue device.
/usr/sbin/mkvirprt	Creates a virtual printer.

Related Information

The **piobe** command, **piomkapqd** command.

Printer Overview for System Management in *AIX 5L Version 5.2 Guide to Printers and Printing*

Spooler Overview for System Management in *AIX 5L Version 5.2 Guide to Printers and Printing*

Printer Backend Overview for Programming in *AIX 5L Version 5.2 Guide to Printers and Printing*

---

piomsg Command

Purpose

Sends a printer backend message to the user.

Syntax

piomsg [**-u** *UserList*] [**-c** *MsgCatalog* [**-s** *MsgSet*] **-n** *MsgNumber*] [**-a** *MsgArg*] ...
[*MsgText*]

Description

The **piomsg** command either retrieves a printer backend message from a message catalog or sends a specified message text to one or more users. The **piomsg** command runs when a print job is executed. Typically, the **piomsg** command is used in printer colon files to send a message to the user submitting a print job while the print job is processed by the **piobe** command.

When the **-c**, **-s**, or **-n** flags are specified, the **piomsg** command retrieves a message from a message catalog. The command searches for the message in the directory specified in the **NLSPATH** environment variable. If the **NLSPATH** environment variable does not contain a directory path, the **piomsg** command searches the **/usr/lib/lpd/pio/etc** default directory. If no message is found in the **/usr/lib/lpd/pio/etc** directory, the command supplies the text specified in the *MessageText* variable. When the **-c**, **-s**, or **-n** flags are not specified, the **piomsg** command returns the value (if any) of the *MessageText* variable.

Each message is parsed for the **%s** or **%n$s printf** subroutine conversion specifications. The **printf** conversion specifications are replaced with supplied message strings, if any, before the message is sent to the user. The **piomsg** command processes escape sequences, such as, linefeed **/n** or horizontal tab **/t**, that are embedded in the message.

Flags

-a *MsgArg*
: Specifies the message argument string. The value of the *MsgArg* variable is substituted into the message, if it contains the **%s** or **%n$s printf** subroutine conversion specifications. The **-a** flag can be specified up to 10 times to specify multiple arguments. If there are any errors while parsing conversion specifications, the original message is sent.

-c *MsgCatalog* Specifies the message catalog that contains the message to be retrieved. The **-c** flag must be specified with the **-n** flag.

-n *MsgNumber* Specifies the message number. The **-n** flag must be specified with the **-c** flag.

-s *MsgSet* Specifies an optional message set. The default value for the *MsgSet* variable is 1. The **-s** flag must be specified with both the **-c** and **-n** flags.

-u *UserList* Specifies the list of users who receive the message. The names of users or nodes in the *UserList* variable are separated by commas. To include a node name in the user list specify the @ character followed by a node name or address. If the **-u** flag is omitted, the message returns to the user who initiated the print job.

Examples

1. To retrieve message number `100` in message set number 1 from the `piobe.cat` message catalog and send the message to user `joe` on the same node as the print server and `tom` on node `foobar`, enter:
   ```
   piomsg -u joe,tom@foobar -c piobe.cat -n 100
   ```
2. To send a message with a message argument string to the user who submitted the print job, enter:
   ```
   piomsg -a "/usr/bin/troff" "The specified filter %s is not found\n"
   ```
3. To retrieve message number 5 in set number2 from the `xyz.cat`, use a dummy message in the event of a failure, and send the message to the printer, enter:
   ```
   piomsg -cxyz.cat -s2 -n5 "xyz.cat is not installed.\n"
   ```

Note: When the **piomsg** command cannot retrieve messages from the catalog specified with the **NLSPATH** environment variable or the default directory, the supplied message text is sent to the users.

File

/usr/lib/lpd/pio/etc/piomsg Contains the **piomsg** command.

Related Information

The **piobe** command.

The **printf** subroutine.

Printer Overview for System Management in *AIX 5L Version 5.2 Guide to Printers and Printing* .

Spooler Overview for System Management in *AIX 5L Version 5.2 Guide to Printers and Printing* .

Printer Backend Overview for Programming in *AIX 5L Version 5.2 Guide to Printers and Printing* .

---

pioout Command

Purpose

Printer backend's device driver interface program.

Syntax

/usr/lpd/pio/etc/pioout [**-A** *BytesPrinted*] [**-B** *TotalBytes*] [**-C** *NumberCancelStrings*] [**-D** *CancelString*] [**-E** *Mask*] [**-F** *FormFeedString*] [**-I** *InterventionRequiredUser*] [**-K** *TextString*] [**-L** *TextString*] [**-N** *NumberFormFeedStrings*] [**-O** *OutFile*] [**-P** *PrefixFile*] [**-R** *ParseRoutine*] [**-S** *SuffixFile*] [**-W+**]

Description

The **pioout** command is at the end of pipelines invoked by the **piobe** command (the print job manager) to print a file or a burst page on a printer. It reads input data from standard input, the prefix file (if the **-P** flag is specified), and the suffix file (if the **-S** flag is specified), and then writes the data to the printer (or *OutFile*, if the **-O** flag is specified). Error conditions and situations where intervention is required (unless the **-I** flag is specified) are reported to the user who submitted the print job.

The values specified with the **-A** flag and the **-B** flag are used to periodically report to the **qdaemon** process the percentage of the print job that has completed. The **-C** flag and the **-D** flag specify the data string sent to the printer if the print job is canceled.

The **-O** flag is used to generate a header page and store it in a temporary file. The **-P** flag is then used to print the header page (that was saved in a temporary file) just prior to printing the print file.

The **pioout** command requires the following environment variables to be initialized:

PIOTITLE	Title of the print job
PIODEVNAME	Device name
PIOQNAME	Print queue name
PIOQDNAME	Queue device name
PIOFROM	User who submitted the print job
PIOMAILONLY	If nonzero, message to user should always be mailed, not displayed.
PIOTERM	Overrides the terminal type assumed from the tty definition. This variable is only used for print jobs submitted to terminal-attached terminals.

Flags

-A *BytesPrinted*	Specifies the number of bytes already printed for the print job.
-B *TotalBytes*	Specifies the total number of bytes to be printed for the print job.
-C *NumberCancelStrings*	Specifies the number of times the string specified by the **-D** flag is to be sent to the printer when a print job is canceled. If this flag is not specified, the value is assumed to be 3168.
-D *CancelString*	Specifies the string to be sent to the printer when a print job is canceled. If the **-D** flag is not specified, the string is assumed to consist of 1 null character.
-E *Mask*	Specifies, as *Mask*, one or more device-driver error-flag names, separated by commas. If the mask is one returned by the **ioctl** subroutine with an **LPQUERY** command, the error condition indicated by the mask is ignored. Flag names can include **LPST_ERROR, LPST_NOSLCT**, and **LPST_SOFT**, and are defined in the **/usr/include/sys/lpio.h** file.
-F *FormFeed String*	Specifies the string to be sent to the printer to cause a form feed. If the **-F** flag is not specified, the string is assumed to be `\014`.
-I *InterventionRequiredUser*	Specifies the user to whom a message is to be sent when the printer requires intervention. If this flag is not specified, the message is sent to the user who submitted the print job. The *InterventionRequiredUser* parameter can be one or more user names, separated by commas. A null string represents the print job submitter. For example, the string **,**`jim@server02` causes intervention required messages to be sent to both the print job submitter and to user `jim` at node `server02`.
-K *TextString*	Specifies that messages sent by a PostScript printer will be discarded if they contain the specified text string. For example, if the *TextString* variable is `warming up`, messages that include the text `warming up` will be discarded.
-L *TextString*	Specifies that if a message received from a PostScript printer includes the specified text string, the text following this text string in the message will be sent to the intervention-required user specified by the **-I** flag.

-N *NumberFormFeedStrings*	Specifies the number of form-feed strings to be sent to the printer at the end of the input data stream. If this flag is not specified, the value is assumed to be zero. This flag is normally used only to align continuous forms after the printer has been idle, or to feed forms when the printer goes idle.
-O *OutFile*	Specifies that the output is sent to the specified file instead of being sent to the printer.
-P *PrefixFile*	Specifies the file sent to the printer before the first byte of the print file is sent. If the print job terminates before the first byte of the print file arrives, the prefix file is not sent.
-R *ParseRoutine*	Specifies the full path name of a routine to parse data read from the printer. An example of a parse routine is contained in the **/usr/include/piostruct.h** file. If the **-R** flag is not specified, a default parse routine is used.
-S *SuffixFile*	Specifies the file sent to the printer after the print file has been sent. If the print job terminates before the first byte of the print file arrives, the suffix file is not sent.
-W +	Specifies that EOF (hex 04) must be received from the printer in order to exit.

Related Information

The **piobe** command, **pioburst** command, **piodigest** command, **pioformat** command, **piofquote** command, **piopredef** command, **qdaemon** command.

Printer Backend Overview for Programming in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

Printer Addition Management Subsystem: Programming Overview in *AIX 5L Version 5.2 Kernel Extensions and Device Support Programming Concepts*.

Printer Code Page Translation Tables in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

Printer Colon File Conventions in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

Printer Colon File Escape Sequences in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

Understanding the Interaction between qdaemon and the Backend in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

Understanding Backend Routines in libqb in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

How to Add a Printer Using the Printer Colon File in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

---

piopredef Command

Purpose

Creates a predefined printer data-stream definition.

Syntax

piopredef [**-r**] **-d** *QueueDeviceName* **-q** *PrintQueueName* **-s** *DataStreamType* **-t** *PrinterType*

Description

The **piopredef** command creates a predefined printer data-stream definition from a virtual printer definition. It can be thought of as the inverse of the **mkvirprt** command. The **mkvirprt** command copies a predefined printer data stream definition to create a virtual printer definition, which can then be customized as desired. The **piopredef** command, however, copies a customized virtual printer definition to create a predefined printer data stream definition.

The **piopredef** command can be used to create a predefined printer definition for an unsupported printer that accepts a print data stream similar to that of a supported printer. For example, a virtual printer definition for a 4201-3 Proprinter III can be created with the **mkvirprt** command, displayed with the **lsvirprt** command, modified as necessary for the unsupported printer with the **chvirprt** command, and then specified with the **piopredef** command to create a predefined definition for the unsupported printer.

The new predefined printer definition can then be specified with a **mkvirprt** command to generate additional virtual printers for the unsupported printer type on the same computer, or transported to other computers and used there.

Flags

-d *QueueDeviceName*
: Specifies with the *QueueDeviceName* variable the spooler of the customized virtual printer definition to be used to create the predefined printer definition.

-q *PrintQueueName*
: Specifies with the *PrintQueueName* variable the spooler of the virtual printer definition to be used to create the predefined printer definition.

-r
: Specifies that if the **-s** flag and the **-t** flag specify a predefined printer definition that already exists, the existing one should be replaced.

-s *DataStreamType*
: Specifies with the *DataStreamType* variable the printer for the predefined printer definition to be created. Example data stream types are:

  asc IBM extended ASCII

  gl Hewlett-Packard GL

  pcl Hewlett-Packard PCL

  ps PostScript

  630 Diablo 630

  855 Texas Instruments 855.

-t *PrinterType*
: Specifies the printer type for the predefined printer definition to be created. Examples of existing printer types are: 4201-3, hplj-2, ti2115, and so on.

Note: If no flags are specified, the command syntax is displayed.

Example

To create a new predefined printer definition from an existing virtual printer definition for the virtual printer, enter:

```
piopredef -d mypro -q proq -s asc -t 9234-2
```

The attributes for the virtual printer assigned to the `mypro` queue device on the `proq` print queue are copied to create a new predefined printer definition for the `9234-2` printer (`asc` data stream).

Files

/etc/piopredef
: Contains the **piopredef** command.

/usr/lpd/pio/predef/*
: Predefined printer data stream attribute files. File names are in the format: `PrinterType.DataStreamType`.

/var/spool/lpd/pio/@local/custom/* Customized virtual printer attribute files. File names are in the format: `PrintQueueName:QueueDeviceName`.

Related Information

Printer Overview for System Management, Printer Specific Information, Virtual Printer Definitions and Attributes, Printer Backend Overview for Programming, Adding a Printer Using the Printer Colon File, Printer Code Page Translation Tables in *AIX 5L Version 5.2 Guide to Printers and Printing*.

Printer Addition Management Subsystem: Programming Overview in *AIX 5L Version 5.2 Kernel Extensions and Device Support Programming Concepts*.

---

pkgadd Command

Purpose

Transfers a software package or set to the system.

Syntax

To Install a Software Package

pkgadd [**-d** *Device*] [**-r** *Response*] [**-n**] [**-a** *Admin*] [*Pkginst1* [*Pkginst2* [. . .]]]

To Copy a Software Package to the Specified Spool Directory

pkgadd -s *Spool* [**-d** *Device*] [*Pkginst1* [*Pkginst2* [. . .]]]

Description

pkgadd transfers the contents of a software package or set from the distribution medium or directory to install it onto the system. A package is a collection of related files and executables that can be independently installed. A set is made up of a special-purpose package, referred to as a Set Installation Package (SIP), and a collection of one or more packages that are members of the set. The SIP controls the installation of the set.

pkgadd checks that all packages listed on the command line are on the installation medium. If any of the packages listed does not exist, no changes are made to the system, that is, none of the listed packages are installed.

Note: You must be ″root″ to install packages. Non-root users can run **pkgadd**, but only up to a certain point.

Used without the **-d** flag, **pkgadd** looks in the default spool directory for the package (**/var/spool/pkg**). Used with the **-s** flag, it writes the package to a spool directory instead of installing it.

Error messages are always logged. In addition, when **pkgadd** terminates, it sends mail (by default, to ″root″) with all the error messages and a summary of which packages installed completely, partially, or not at all.

Flags

-d *Device* Installs or copies a package/set from *Device*. *Device* can be the full pathname to a directory, file or named pipe, or ″-″ which specifies packages in datastream format read from standard input. The default device is the installation spool directory (**/var/spool/pkg**).

-r *Response* Identifies a file or directory, *Response*, which contains the answers to questions posed by a ″request script″ during a previous **pkgask** session conducted in interactive mode [see the **pkgask** command]. When *Pkginst* is a package, *Response* can be a full pathname or a directory; when *Pkginst* is a SIP, *Response* must be a directory.

-n Specifies that installation runs in non-interactive mode. The default mode is interactive.

-a *Admin* Defines an installation administration file, *Admin*, to be used in place of the default administration file to specify whether installation checks (such as the check on the amount of space, the system state, and so on) are done. The token ″none″ overrides the use of any **admin** file, and thus forces interaction with the user. Unless a full pathname is given, **pkgadd** looks in the **/var/sadm/install/admin** directory for the file. By default, the file **default** in that directory is used. **default** specifies that no checking is done, except to see if there is enough room to install the package and if there are dependencies on other packages. The **-a** flag cannot be used if *Pkginst* is a SIP.

Pkginst Defines a short string used to designate an abbreviation for the package/set name. (The term ″package instance″ is used loosely: it refers to all instantiations of *Pkginst*.) See the **pkginfo** command and the **pkginfo** file format.

If *Pkginst* is a SIP, the SIP controls installation of the set by using request scripts and pre-install scripts. The SIP request script, not the package installation tools, is responsible for prompting the user for responses and taking the appropriate actions. If the request script fails, only the SIP is processed.

To indicate all instances of a package, specify '*Pkginst***.***', enclosing the command line in single quotes, as shown, to prevent the shell from interpreting the ″*″ character. Use the token ″all″ to refer to all packages available on the source medium.

-s *Spool* Reads the package into the directory *Spool* instead of installing it.

Special Notes

The **-r** flag can be used to indicate a directory name as well as a filename. The directory can contain numerous *Response* files, each sharing the name of the package with which it should be associated. This would be used, for example, when adding multiple interactive packages with one invocation of **pkgadd**. Each package that had a request script would need a *Response* file. If you create response files with the same name as the package (for example, *Package1* and *Package2*) then, after the **-r** flag, name the directory in which these files reside.

The **-n** flag causes the installation to halt if any interaction is needed to complete it.

When invoked with no *Pkginst* specified on the command line, **pkgadd** only displays the names of sets if at least one SIP exists on the media. Because of this, you shouldn't include packages on the same media if some are members of sets and some are not. If you do, the packages which are not members of sets can be installed only if their **pkginst** names are provided on the command line.

The **pkgadd** command checks to see if any of the files in *Pkginst* are already installed on the system and, if any are, saves this fact before continuing with installation. Later, **pkgadd** does not reinstall these files on the system. If one of the packages installation scripts removes such a file, the result is that the file will no longer be on the system when package installation completes.

The **pkgadd** command does not uncompress any files that were already compressed (that is, only those in ″**.Z**″ form) before being processed by **pkgmk**.

Exit Status

This command returns the following exit values:

0 Successful completion of script.

1 Fatal error. Installation process is terminated at this point.

2 Warning or possible error condition. Installation continues. A warning message is displayed at the time of completion.

3 Script was interrupted and possibly left unfinished. Installation terminates at this point.
4 Script was suspended (administration). Installation terminates at this point.
5 Script was suspended (interaction was required). Installation terminates at this point.
10 System should be rebooted when installation of all selected packages is completed. (This value should be added to one of the single-digit exit codes described above.)
20 The system should be rebooted immediately upon completing installation of the current package. (This value should be added to one of the single-digit exit codes described above.)
77 No package was selected for the set.
99 Internal error.

Files

/var/sadm/install/admin/default	default package administration file
/var/sadm/install/logs/*pkginst*.log	error message log
/var/spool/pkg	default spool directory

Related Information

The **pkgask** command, **pkgchk** command, **pkginfo** command, **pkgmk** command, **pkgparam** command, **pkgrm** command.

---

pkgask Command

Purpose

Stores answers to a request script.

Syntax

pkgask [**-d** *Device*] **-r** *Response* [*Pkginst* [*Pkginst* [. . .]]

Description

pkgask enables an administrator to store answers to an interactive package (one with a request script) or a set of packages. A set is made up of a special-purpose package, referred to as a Set Installation Package (SIP), and a collection of one or more packages that are members of the set. The SIP controls the installation of the set.

Invoking **pkgask** generates a *Response* file that is then used as input at installation time. The use of this *Response* file prevents any interaction from occurring during installation since the file already contains all of the information the package needs.

When **pkgask** runs, it creates the response file as well as the following directories:

/ptfvars	Contains variables pertaining to the package.
/fileinfo	Contains checksum information about the package.
/oldfiles	Contains backups of previous versions of the package.

To install the package on another system non-interactively, you must copy all of these files and directories to the target system.

Note: If you overwrite any of these directories, for example, to install another package non-interactively, you will not be able to successfully remove the first package unless you restore the original directory contents first.

You can use the **-r** flag to indicate a directory name as well as a filename. The directory name is used to create numerous *Response* files, each sharing the name of the package with which it should be associated. This is useful, for example, when you add multiple interactive packages with one invocation of **pkgadd**. Each package needs a *Response* file. To create multiple response files with the same name as the package instance, name the directory in which the files should be created and supply multiple instance names with the **pkgask** command. When installing the packages, you can identify this directory to the **pkgadd** command.

Flags

-d *Device*
: Runs the request script for a package on *Device*. *Device* can be the full pathname to a directory (such as **/var/tmp**), or ″-″ which specifies packages in datastream format read from standard input. The default device is the installation spool directory (**/var/spool/pkg**).

-r *Response*
: Identifies a file or directory, *Response*, which should be created to contain the responses to interactions with the packages request script. The file, or directory of files, can later be used as input to the **pkgadd** command [see the **pkgadd** command]. When *Pkginst* is a package, *Response* can be a full pathname or a directory; when *Pkginst* is a SIP, *Response* must be a directory.

Pkginst
: Defines a short string used to designate an abbreviated package/set name. (The term ″package instance″ is used loosely: it refers to all instantiations of *Pkginst*, even those that do not include instance identifiers.)

  To create a package name abbreviation, assign it with the ″PKG″ parameter. For example, to assign the abbreviation ″cmds″ to the Advanced Commands package, enter **PKG=cmds**.

  If *Pkginst* specifies a SIP, all request scripts for packages which are members of that set are run (if any) and the resulting response files are placed in the directory provided to the **-r** flag.

  To indicate all instances of a package, specify '*Pkginst***.***', enclosing the command line in single quotes, as shown, to prevent the shell from interpreting the ″*″ character. Use the token ″all″ to refer to all packages available on the source medium.

  > **Note:** When invoked with no *Pkginst* specified on the command line, **pkgask** only displays the names of sets if at least one SIP exists on the device. Thus, if you have packages which are not members of sets, they can be referenced only if their *Pkginst* names are provided on the command line.

Exit Status

This command returns the following exit values:

0 Successful completion of script.

1 Fatal error. Installation process is terminated at this point.

2 Warning or possible error condition. Installation continues. A warning message is displayed at the time of completion.

3 Script was interrupted and possibly left unfinished. Installation terminates at this point.

4 Script was suspended (administration). Installation terminates at this point.

5 Script was suspended (interaction was required). Installation terminates at this point.

10 System should be rebooted when installation of all selected packages is completed. (This value should be added to one of the single-digit exit codes described above.)

20 The system should be rebooted immediately upon completing installation of the current package. (This value should be added to one of the single-digit exit codes described above.)

77 No package was selected for the set.

99 Internal error.

Files

/var/spool/pkg	default spool directory

Related Information

The **pkgadd** command, **pkgchk** command, **pkginfo** command, **pkgmk** command, **pkgparam** command, **pkgtrans** command, **pkgrm** command.

---

pkgchk Command

Purpose

Checks the accuracy of an installation.

Syntax

To Check the Contents of Installed Objects

pkgchk [**-l** | **-a -c -f -q -v**] [**-n -x**] [**-p** *Path1*[,*Path2* . . .] [**-i** *File*] [*Pkginst* . . .]

To Check the Contents of a Package Spooled on a Specified Device

pkgchk -d *Device* [**-l** | **-v**] [**-p** *Path1*[,*Path2* . . .] [**-i** *File*] [*Pkginst* . . .]

To Check the Contents of a Package Described in the Specified pkgmap

pkgchk -m *Pkgmap* [**-e** *Envfile*] [**-l** | **-a -c -f -q -v**] [**-n -x**] [**-i** *File*] [**-p** *Path1*[,*Path2* . . .]]

Description

pkgchk checks the accuracy of installed files or, by use of the **-l** flag, displays information about package files. The command checks the integrity of directory structures and the files. Discrepancies are reported on **stderr** along with a detailed explanation of the problem.

The first synopsis defined above is used to list or check the contents and/or attributes of objects that are currently installed on the system. Package names can be listed on the command line, or by default the entire contents of a machine is checked.

The second synopsis is used to list or check the contents of a package which has been spooled on the specified device, but not installed. Note that attributes cannot be checked for spooled packages.

The third synopsis is used to list or check the contents and/or attributes of objects which are described in the indicated *Pkgmap*.

Flags

-l	Lists information on the selected files that make up a package. It is not compatible with the **a**, **c**, **f**, **g**, and **v** flags.
-a	Audits the file attributes only, does not check file contents. Default is to check both.
-c	Audits the file contents only, does not check file attributes. Default is to check both.
-f	Corrects file attributes if possible. If used with the **-x** flag, it removes hidden files. When **pkgchk** is invoked with this flag it creates directories, named pipes, links, and special devices if they do not already exist.
-q	Enables quiet mode. Does not give messages about missing files.
-v	Enables verbose mode. Files are listed as processed.
-n	Ignores volatile or editable files. This should be used for most post-installation checking.

-x Searches exclusive directories only, looking for files that exist that are not in the installation software database or the indicated *Pkgmap* file. (An exclusive directory is a directory created by and for a package; it should contain only files delivered with a package. If any non-package files are found in an exclusive directory, **pkgchk** reports an error.) If **-x** is used with the **-f** flag, hidden files are removed; no other checking is done.

> **Note:** To remove hidden files only, use the **-f** and **-x** flags together. To remove hidden files and check attributes and contents of files, use the **-f**, **-x**, **-c**, and **-a** flags together.

-p Only checks the accuracy of the pathname or pathnames listed. ″pathname″ can be one or more pathnames separated by commas (or by white space, if the list is quoted).

-i Reads a list of pathnames from *File* and compares this list against the installation software database or the indicated *Pkgmap* file. Pathnames that are not contained in ″inputfile″ are not checked.

-d Specifies the device on which a spooled package resides. *Device* can be a directory pathname, or ″-″ which specifies packages in datastream format read from standard input.

-m Requests that the package be checked against the pkgmap file *Pkgmap*.

-e Requests that the pkginfo file named as *Envfile* be used to resolve parameters noted in the specified pkgmap file.

Pkginst Defines a short string used to designate an abbreviation for the package name. (The term ″package instance″ is used loosely: it refers to all instantiations of *Pkginst*, even those that do not include instance identifiers.)

To indicate all instances of a package, specify '*Pkginst*.*****', enclosing the command line in single quotes, as shown, to prevent the shell from interpreting the ″*″ character. Use the token ″all″ to refer to all packages available on the source medium.

Exit Status

This command returns the following exit values:

0 Successful completion of script.

1 Fatal error. Installation process is terminated at this point.

Files

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

Related Information

The **pkgadd** command, **pkgask** command, **pkginfo** command, **pkgmk** command, **pkgrm** command, **pkgtrans** command.

The **pkginfo** file format, **pkgmap** file format.

---

pkginfo Command

Purpose

Displays software package and/or set information.

Syntax

To Display Information on Installed Packages

pkginfo [**-q**] [**-x** | **-l**] [**-r**] [**-p** | **-i**] [**-a** *Arch*] [**-v** *Version*] [**-c** *Category1*,[*Category2*[, . . .]]] [*Pkginst* [, *Pkginst* [, . . .]]]

To Display Information on Packages Contained in the Specified Device

pkginfo [**-d** *Device*] [**-q**] [**-x** | **-l**] [**-a** *Arch*] [**-v** *Version*] [**-c** *Category1* [,*Category2*[, . . .]]] [*PkginstPkginst* [, *Pkginst* [, . . .]]]

Description

pkginfo displays information about software packages or sets that are installed on the system (as requested in the first synopsis) or that reside on a directory (as requested in the second synopsis). A package is a collection of related files and executables that can be independently installed. A set is made up of a special-purpose package, referred to as a Set Installation Package (SIP), and a collection of one or more packages that are members of the set. The SIP controls the installation of the set.

When run without flags, **pkginfo** displays one line of information about every installed package (whether installed completely or partially) whose category is not the value ″set″. The information displayed includes the primary category, package instance, and name of the package. For UNIX software packages produced before UNIX System V Release 4, **pkginfo** displays only the package name and abbreviation.

The **-p** and **-i** flags are meaningless if used in conjunction with the **-d** flag. The **-p** and **-i** flags are mutually exclusive. The **-x** and **-l** flags are mutually exclusive.

Flags

-q — Enables quite mode - no information is displayed. This flag overrides the **-x**, **-l**, **-p**, and **-i** flags. (Can be invoked by a program to query whether or not a package has been installed.)

-x — Extracts and displays the following information about the specified package: abbreviation, name, and, if available, architecture and version.

-l — Displays a ″long format″ report (that is, one that includes all available information) about the specified package(s).

-r — Lists the installation base for the specified package if the package is relocatable.

-p — Displays information only for partially installed packages.

-i — Displays information only for fully installed packages.

-a *Arch* — Specifies the architecture of the package as *Arch*.

-v *Version* — Specifies the version of the package as *Version*. All compatible versions can be requested by preceding the version name with a tilde ″~″.

-c *Category* . . . — Displays information about packages that belong to category *Category*. (Categories are defined in the category field of the **pkginfo** file; see the **pkginfo** file format for details.) More than one category may be specified in a comma-separated list. A package is required to belong to only one category, even when multiple categories are specified. The package-to-category match is not case-sensitive.

If the category specified is ″set″, **pkginfo** displays information about Set Installation Packages (SIPs).

Pkginst — Defines a short string used to designate an abbreviation for the package/set name. (The term ″package instance″ is used loosely: it refers to all instantiations of *Pkginst*, even those that do not include instance identifiers.)

To indicate all instances of a package, specify '*Pkginst***.***', enclosing the command line in single quotes, as shown, to prevent the shell from interpreting the ″*″ character. Use the token ″all″ to refer to all packages available on the source medium.

If *Pkginst* is a SIP, information about the packages with which the SIP is associated is displayed.

-d *Device* — Displays information from packages/sets that reside on *Device*. *Device* can be the full pathname to a directory (such as **/var/tmp**), or ″-″ which specifies packages in datastream format read from standard input. The default device is the installation spool directory (**/var/spool/pkg**).

Exit Status

This command returns the following exit values:

0 Successful completion of script.
1 Fatal error. Installation process is terminated at this point.

Files

/var/spool/pkg Default spool directory

Related Information

The **pkgadd** command, **pkgask** command, **pkgchk** command, **pkgmk** command, **pkgrm** command, **pkgtrans** command.

The **pkginfo** file format, **setinfo** file format.

---

pkgmk Command

Purpose

Produces an installable package.

Syntax

pkgmk [**-c**] [**-o**] [**-a** *Arch*] [**-b** *BaseDir*] [**-d** *Directory*] [**-f** *Prototype*] [**-l** *Limit*] [**-p** *PStamp*] [**-r** *RootPath*] [**-v** *Version*] [*Variable=Value ...*] [*PkgInst*]

Description

pkgmk produces an installable package to be used as input to the **pkgadd** command. A package is a collection of related files and executables that can be independently installed. The package contents will be in directory structure format.

The **pkgmk** command uses the package prototype file as input and creates a **pkgmap** file. The contents for each entry in the prototype file is copied to the appropriate output location. Information concerning the contents (checksum, file size, modification date) is computed and stored in the **pkgmap** file, along with attribute information specified in the prototype file.

Flags

-a *Arch* Overrides the architecture information provided in the **pkginfo** file with *Arch*.

-b *BaseDir* Prepends the indicated *BaseDir* to locate relocatable objects on the source machine.

-c Compresses non-information files. You must also specify the **-r** option when using **-c**. Entries in the *Prototype* file that reference relative paths above the *RootPath* specification will not be compressed. Any files that were already compressed (that is, only those in ″.Z″ form) before being processed by **pkgmk** will not be uncompressed by the **pkgadd** command.

-d *Directory* Creates the package in *Directory*. The directory named must already exist.

-f *Prototype*	Uses the file *Prototype* as input to the command. The default name for this file is either **Prototype** or **prototype**. You can use **pkgproto** to create the *Prototype* file. In this case, you must manually add in the entries for any installation scripts and files you are using in the package. You only need entries for those files and scripts that you use. However, you must always add an entry for the **pkginfo** file for the package. See **pkgproto** for more information.
-l *Limit*	Specifies the maximum size in 512-byte blocks of the output device as *Limit*. By default, if the output file is a directory or a mountable device, **pkgmk** will employ the **df** command to dynamically calculate the amount of available space on the output device. Useful in conjunction with **pkgtrans** to create a package with datastream format.
-o	Overwrites the same instance. The package instance will be overwritten if it already exists.
-p *PStamp*	Overrides the production stamp definition in the **pkginfo** file with *PStamp*.
-r *RootPath*	Appends the source pathname in the *Prototype* file to the indicated *RootPath* to locate objects on the source machine.
-v *Version*	Overrides version information provided in the **pkginfo** file with *Version*.
Variable=*Value*	Places the indicated variable in the packaging environment.
PkgInst	A short string used to designate an abbreviation for the package name. **pkgmk** will automatically create a new instance if the version and/or architecture is different. A user should specify only a package abbreviation; a particular instance should not be specified unless the user is overwriting it.

Examples

1. If you want to create a package named `mypkgA` containing the **lsps** and **lsuser** commands, you must first create the contents of the package. For example:

   ```
   mkdir -p /home/myuser/example/pkgmk/sbin
   cp /usr/sbin/lsps /home/myuser/example/pkgmk/sbin
   cp /usr/sbin/lsuser /home/myuser/example/pkgmk/sbin
   ```

   Then, create the **pkginfo** file. In this example the **pkginfo** file is `/home/myuser/example/pkgmk/pkginfo`, which contains the following:

   ```
   PKG="mypkgA"
   NAME="My Package A"
   ARCH="PPC"
   RELEASE="1.0"
   VERSION="2"
   CATEGORY="Application"
   PSTAMP="AIX  2001/02/05"
   ```

   Then, create the *Prototype* file, `/home/myuser/example/pkgmk/prototype` file which contains the following:

   ```
   !search /home/myuser/example/pkgmk/sbin
   i pkginfo=/home/myuser/example/pkgmk/pkginfo
   d example /example 1777 bin bin
   ```

```
d example /example/pkgmk 1777 bin bin
d example /example/pkgmk/sbin 1777 bin bin
f example /example/pkgmk/sbin/lsps 555 bin bin
f example /example/pkgmk/sbin/lsuser 555 bin bin
```

Then, create the package with the above *Prototype* and **pkginfo** files using the **pkgmk** command:

```
pkgmk -d /tmp -f /home/myuser/example/pkgmk/prototype
```

This produces the following output:

```
Building pkgmap from package prototype file
## Processing pkginfo file
    WARNING:parameter <CLASSES> set to "example"

## Attempting to volumize 5 entries in pkgmap
Part  1 -- 218 blocks, 10 entries
/tmp/mypkgA/pkgmap
/tmp/mypkgA/pkginfo
/tmp/mypkgA/root/example/pkgmk/sbin/lsps
/tmp/mypkgA/root/example/pkgmk/sbin/lsuser
## Packaging complete
```

The newly created package named `mypkgA` now exists in `/tmp/mypkgA`.

Exit Status

0	Successful completion of script.
1	Fatal error. Installation process is terminated at this point.
99	Internal error.

Files

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

Related Information

The **installp** command, **pkgadd** command, **pkgask** command, **pkgchk** command, **pkginfo** command, **pkgparam** command, **pkgproto** command, **pkgrm** command, **pkgtrans** command, **tar** command.

The **pkginfo** file format.

For information about preparing applications to be installed using the **installp** command, refer to Packaging Software for Installation in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

---

pkgparam Command

Purpose

Displays package parameter values.

Syntax

To Display the Value of a Parameter Contained in pkginfo

pkgparam [**-v**] [**-d** *Device*] *Pkginst* [*Param* **...**]

To Display the Value of a Parameter Contained in a Device

pkgparam -d *Device* [**-v**] [*Param* **...**]

To Display the Value of a Parameter Contained in a File

pkgparam -f *File* [**-v**] [*Param* **...**]

Description

pkgparam displays the value associated with the parameter or parameters requested on the command line. The values are located in one of the following places: in the **pkginfo** file for *Pkginst*, on the *Device* named with the **-d** flag, or on the specific file named with the **-f** flag. When a *Device* is given, but a *Pkginst* is not (as shown in the second synopsis), parameter information for all packages residing on *Device* is shown.

One parameter value is shown per line. Only the value of a parameter is given unless the **-v** flag is used. With this flag, the output of the command is in this format:

```
Parameter1='Value1'
Parameter2='Value2'
Parameter3='Value3'
```

If no parameters are specified on the command line, values for all parameters associated with the package are shown.

Flags

-v	Specifies verbose mode. Displays name of parameter and its value.
-d *Device*	Specifies the *Device* on which a *Pkginst* is stored. *Device* can be the full pathname to a directory (such as **/var/tmp**), or ″-″ which specifies packages in datastream format read from standard input.
-f	Requests that the command read *File* for parameter values. This file should be in the same format as a **pkginfo** file. As an example, such a file might be created during package development and used while testing software during this stage.
Pkginst	Defines a specific package for which parameter values should be displayed. The format *Pkginst*.* can be used to indicate all instances of a package. When using this format, enclose the command line in single quotes to prevent the shell from interpreting the ″*″ character.
Param	Defines a specific parameter whose value should be displayed.

Exit Status

If parameter information is not available for the indicated package, the command exits with a non-zero status.

0	Successful completion of script.
1	Fatal error. Installation process is terminated at this point.

Files

/var/spool/pkg	default spool directory
/usr/sbin/pkgparam	Contains the **pkgparam** command.

Related Information

The **pkgtrans** command.

The **pkginfo** file format.

pkgproto Command

Purpose

Generates a prototype file.

Syntax

pkgproto [**-i**] [**-c** *Class*] [*Path1* [**=***Path2*] ...]

Description

The **pkgproto** commands scans the indicated paths and generates a prototype file that may be used as input to the **pkgmk** command. To do this, the standard output of this command must be redirected to a file. The file can then be used when invoking **pkgmk**.

If no *Paths* are specified on the command line, standard input is assumed to be a list of *Paths*. If the *Path* listed on the command line is a directory, the contents of the directory are searched. However, if input is read from stdin, a directory specified as a path is not searched.

The prototype file attributes *mac*, *fixed*, and *inherited*, cannot be determined by **pkgproto** and must be manually added to the file.

By default, **pkgproto** creates symbolic link entries for any symbolic link encountered (ftype=s). When you use the **-i** flag, **pkgproto** creates a file entry for symbolic links (ftype=f). The prototype file must be edited to assign file types such as v (volatile), e (editable), or x (exclusive directory). **pkgproto** detects linked files. If multiple files are linked together, the first path encountered is considered the source of the link.

The output from this command is sent to standard output. You must redirect standard output to a file if you wish to use the result as a prototype file when invoking **pkgmk**. Since **pkgmk** uses prototype as the default filename for the prototype file, we suggest you direct the output of **pkgproto** to the file name prototype.

You must add entries to the prototype file produced by this command for any installation scripts and files your package may need. At minimum, you will need an entry for the **pkginfo** file. You may also need entries for any of the following files you use in your package: **copyright**, **compver**, **depend**, **setinfo**, **space**, any installation or removal scripts you define for the package, and/or any classes you define.

Notes:

1. By default, **pkgproto** creates symbolic link entries for any symbolic link encountered (ftype=s). When you use the `-i` option, **pkgproto** creates a file entry for symbolic links (ftype=f). The prototype file must be edited to assign file types such as v (volatile), e (editable), or x (exclusive directory). **pkgproto** detects linked files. If multiple files are linked together, the first path encountered is considered the source of the link.
2. The output from this command is sent to standard output. You must redirect standard output to a file if you wish to use the result as a prototype file when invoking **pkgmk**. Since **pkgmk** uses prototype as the default filename for the prototype file, we suggest you direct the output of **pkgproto** to the file name **prototype**.
3. Note that you must add entries to the **prototype** file produced by this command for any installation scripts and files your package may need. At minimum, you will need an entry for the **pkginfo** file; see **pkginfo** for more information. You may also need entries for any of the following files you use in your package: **copyright**, **compver**, **depend**, **setinfo**, **space**, any installation or removal scripts you define for the package, and/or any classes you define, (e.g., postinstall).

Flags

-i	Ignores symbolic links and records the paths as ftype=f (a file) versus ftype=s (symbolic link).
-c *Class*	Maps the class of all paths to *Class*.
Path1	Path of directory where objects are located.
Path2	Path that should be substituted on output for *Path1*.

Examples

The following examples show uses of **pkgproto** and a partial listing of the output produced.

1.
```
$ pkgproto /usr/bin=bin /usr/usr/bin=usrbin /etc=etc
f none bin/sed=/bin/sed 0775 bin bin
f none bin/sh=/bin/sh 0755 bin daemon
f none bin/sort=/bin/sort 0755 bin bin
d none etc/master.d 0755 root daemon
f none etc/master.d/kernel=/etc/master.d/kernel 0644 root daemon
f none etc/rc=/etc/rc 0744 root daemon
```
2.
```
$ find / -type d -print | pkgproto
d none / 755 root root
d none /usr/bin 755 bin bin
d none /usr 755 root root
d none /usr/bin 775 bin bin
d none /etc 755 root root
d none /tmp 777 root root
```
3.

   Identical to the previous example, but with the output captured in a file for later processing with **pkgmk**. Entries added for the required **pkginfo** file, and, for instance, a postinstall script that might be executed after the files are copied into the correct locations.

```
$ find / -type d -print | pkgproto >prototype
$ (edit the file to add entries for pkginfo and postinstall)
$ cat prototype
i pkginfo
i postinstall
d none / 755 root root
d none /usr/bin 755 bin bin
d none /usr 755 root root
d none /usr/bin 775 bin bin
d none /etc 755 root root
d none /tmp 777 root root
```

Return Codes

0	Successful completion of script.
1	Fatal error. Installation process is terminated at this point.

Files

/usr/sbin/pkgproto	Contains the **pkgproto** command

Related Information

The **pkgmk** command, **pkgparam** command, **pkgtrans** command.

The **pkginfo** file

pkgrm Command

Purpose

Removes a package or set from the system.

Syntax

To Remove an Installed Software Package

pkgrm [**-n**] [**-a** *Admin*] [*Pkginst1* [*Pkginst2* [. . .]]]

To Remove a Software Package from a Spool Device

pkgrm -s *Spool* [*Pkginst*]

Description

pkgrm removes a previously installed or partially installed package/set from the system. A package is a collection of related files and executables that can be independently installed. A set is made up of a special-purpose package, referred to as a Set Installation Package (SIP), and a collection of one or more packages that are members of the set.

pkgrm checks that all packages listed on the command line are on the system. If any of the packages listed does not exist, no changes are made to the system, that is, none of the listed packages are removed.

A check is also made to determine if any other packages depend on the one being removed. The action taken if a dependency exists is defined in the *Admin* file (see the **-a** flag, below).

The default state for the command is interactive mode, meaning that prompt messages are given during processing to allow the administrator to confirm the actions being taken. Non-interactive mode can be requested with the **-n** flag.

The **-s** flag can be used to specify the directory from which spooled packages should be removed.

Flags

-n Enables non-interactive mode. If there is a need for interaction, the command exits. Use of this flag requires that at least one package instance be named upon invocation of the command.

-a *Admin* Defines an installation administration file, *Admin*, to be used in place of the default administration file. [For a description of the format of an *Admin* file, see the **admin** file format.] The token ″none″ overrides the use of any *Admin* file, and thus forces interaction with the user. Unless a full pathname is given, **pkgrm** looks in the **/var/sadm/install/admin** directory for the file. By default, the file **default** in that directory is used.

-s *Spool* Removes the specified package(s) from the directory *Spool*.

Pkginst Defines a short string used to designate an abbreviation for the package/set name. (The term ″package instance″ is used loosely: it refers to all instantiations of *Pkginst*, even those that do not include instance identifiers.)

If *Pkginst* specifies a SIP, all installed packages which are members of the set, and the SIP itself, are removed in reverse dependency order.

To indicate all instances of a package, specify '*Pkginst***.***'**, enclosing the command line in single quotes, as shown, to prevent the shell from interpreting the ″*″ character. Use the token ″all″ to refer to all packages available on the source medium.

Exit Status

This command returns the following exit values:

0 Successful completion of script.
1 Fatal error. Installation process is terminated at this point.
99 Internal error.

Files

/usr/sbin/pkgrm	Contains the **pkgrm** command

Related Information

The **pkgadd** command, **pkgask** command, **pkgchk** command, **pkginfo** command, **pkgmk** command, **pkgparam** command, **pkgtrans**command.

The **pkginfo** file format, **pkgmap** file format.

---

pkgtrans Command

Purpose

Translates package format.

Syntax

pkgtrans [**-i -o -n -s**] [**-z** *Blocksize*] *Device1 Device2* [*Pkginst1* [*Pkginst2* [...]]]

Description

The **pkgtrans** command translates an installable package from one format to another. It translates the following:

- A file system format to a datastream
- A datastream to a file system format

You cannot run **pkgtrans** from **csh**.

Flags

-i	Copies the **pkginfo** and *Pkgmap* files. If the packages category is defined as ″set″ for Set Installation Packages (SIPs) (see **setinfo** file format), then that packages' **setinfo** file is also copied.
-o	Overwrites the same instance on the destination device. The package instance is overwritten if it already exists.
-n	Creates a new instance of the package on the destination device. If the package instance already exists on the destination device, it is left unchanged and a new instance is created. The new instance has a sequence number attached to distinguish it from the existing instance. For example, assume the destination device already contained an instance of package *X*. If you use **pkgtrans** with the **-n** flag to write a new instance of package *X* to the device, the existing instance of package *X* remains on the destination device, and a new instance, called *X.2*, would be created on the device. If you executed **pkgtrans** again with the **-n** flag, a third instance, called *X.3*, would be created.
-s	Indicates that the package should be written to *Device2* as a datastream rather than as a file system. The default behavior is to write to *Device2* in the file system format.

-z *Blocksize*	Indicates the blocksize to be used when transferring to cartridge tape. Packages that have been written to tape using the **-z** flag and a value not equal to 512 are always read using a blocksize of 32768. Thus, the **-z** flag is not applicable when reading from cartridge tape.
Device1	Indicates the source device. Can be - (hyphen) which specifies packages in datastream format read from standard input. The package or packages on this device are translated and placed on *Device2*. If *Device1* is a regular file or directory, you must use the absolute pathname, rather than a relative pathname.
Device2	Indicates the destination device. Can be - (hyphen) which specifies packages written to standard output in datastream format. Translated packages are placed on this device. If *Device2* is a regular file or directory, you must specify it as an absolute pathname, rather than a relative pathname.
Pkginst	Specifies which package on *Device1* should be translated. The token ″all″ may be used to indicate all packages. *Pkginst.** can be used to indicate all instances of a package. If no packages are defined, a prompt shows all packages on the device and asks which to translate. If a set is being transferred to datastream format, the *Pkginst* arguments should begin with the SIP and be followed by the packages listed in the SIP's **setinfo** file, in the order in which they appear in that file.

Note: By default, **pkgtrans** does not transfer any instance of a package if any instance of that package already exists on the destination device. Use of the **-n** flag creates a new instance if an instance of this package already exists. Use of the **-o** flag overwrites the same instance if it already exists. Neither of these flags are useful if the destination device is a datastream, because the entire datastream is overwritten anyway.

Exit Status

This command returns the following exit values:

0	Successful completion of script.
1	Fatal error. Installation process is terminated at this point.

Files

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

Examples

1. To translate all packages located on drive *Device* and place the translations in **/tmp**, type:
   ```
   pkgtrans Device /tmp all
   ```
2. To translate ″pkg1″ and ″pkg2″ in **tmp** and place them on *Device* in a datastream format, type:
   ```
   pkgtrans -s /tmp Device pkg1 pkg2
   ```

Related Information

The **pkgadd** command, **pkgask** command, **pkginfo** command, **pkgmk** command, **pkgparam** command, **pkgrm** command.

The **pkginfo** File Format.

---

plotgbe Command

Purpose

Plots HP-GL files to a plotter device.

Syntax

/usr/lpd/plotgbe [**-fr=***X*] [**-noin**] *File*

Description

The **plotgbe** command is a backend program which plots HP-GL files to a plotter device. The plotter device must be attached to a 5085/5086 workstation via the 5080 Attachment Adapter. To use the **plotgbe** command, you must define a print queue for the **plotgbe** backend program. See ″How to Add Plotter Support with 5080″ to learn how to use SMIT commands to add a plotter queue which specifies the **plotbge** backend.

The **plotgbe** command is called by the **qdaemon** process. It should not be entered on the command line. Any options needed for a specific print request to a plotter should be passed to the **plotgbe** command with the command used to request a print job. In the case of the **enq** command, use the **-o** flag to pass options to the **plotgbe** backend for processing.

The **plotgbe** backend command also generates the appropriate HP-GL commands for plotter initialization and plot scaling. This data is sent to the plotter before the user-specified HP-GL file is sent. Thus, any scaling or initialization commands included in the HP-GL file override those generated by the **plotgbe** backend command.

> **Note:** The user must have read access to the file sent to the **plotgbe** command with the print request command.

Flags

-fr=*X* Provides for plotting multi-frame drawings. This option causes *X* number of frames to be plotted, where *X* is a number in the range 1 through 9. For example, plotting a 20’ drawing on E-size role media may require 5 frames. Thus, the option `fr=5` would be passed to the **plotgbe** backend.

-noin Allows plotter front panel settings to remain in effect for the current plot without being reset to default values. Normally, the P1 and P2 positions which define the plot page on the plotter are set by the **plotgbe** command to their default location. Use the **-noin** no-initialization option to override the default locations.

Examples

1. To send the file `longaxis.gl` to the `plt` plotter queue and specify to the backend that the file requires five frames to print, enter:

   ```
   enq -Pplt -o -fr=5 longaxis.gl
   ```

2. To send the file `plotdata.gl` to the `plt` plotter queue, specifying that the plot page positions are not to be reset to default for this file, enter:

   ```
   enq -Pplt -o -noin plotdata.gl
   ```

3. To send the file `twoplot.gl` to the `plt` plotter queue, specifying no plot page initialization and that the plotter print the drawing in two frames, enter:

   ```
   enq -Pplt -o -noin -o fr=2 twoplot.gl
   ```

Files

/usr/lpd/plotgbe Contains the **plotgbe** command.

Related Information

The **enq** command, **plotlbe** command, **qdaemon** command.

Printer Overview for System Management in *AIX 5L Version 5.2 Guide to Printers and Printing* .

Adding Plotter Support with 5080 in *AIX 5L Version 5.2 Guide to Printers and Printing* .

plotlbe Command

Purpose

Plots HP-GL files to a plotter device.

Syntax

/usr/lpd/plotlbe [**-fr=***X*] [**-noin**] *File*

Description

The **plotlbe** command is a backend program which plots HP-GL files to a plotter attached to a serial port defined as a TTY device. To use the **plotlbe** command, you must define a TTY device for the serial port and define a print queue for the **plotlbe** backend program.

When configuring the TTY serial port, set the baud-rate, parity, and stop bits to the appropriate settings for your plotter. You must also set XON/XOFF to `FALSE` for your TTY port.

The **plotlbe** command is called by the **qdaemon** process. It should not be entered on the command line. Any options needed for a specific print request to a plotter should be passed to the **plotlbe** command with the command used to request a print job (usually the **enq** command). With the **enq** command, use the **-o** flag to pass options to the **plotlbe** backend for processing.

The **plotlbe** backend command supports the following plotters: 7731, 7372, 7374, 7375-1, 7375-2, 6180, 6182, 6184, 6186-1, and 6186-2.

The **plotlbe** command supports ENQ/ACK handshaking. Refer to your plotter programming manual for more information on handshaking.

The **plotlbe** backend command also generates the appropriate HP-GL commands for plotter initialization and plot scaling. This data is sent to the plotter before the user-specified HP-GL file is sent. Thus, any scaling or initialization commands included in the HP-GL file override those generated by the **plotlbe** backend command.

> **Note:** The user must have read access to the file sent to the **plotlbe** command with the print request command.

Flags

-fr=*X* Provides for plotting multi-frame drawings. This option causes *X* number of frames to be plotted, where *X* is a number in the range 1 through 9. For example, plotting a 20' drawing on E-size roll media may require 5 frames. Thus, the option `-fr=5` would be passed to the **plotlbe** backend.

-noin Allows plotter front panel settings to remain in effect for the current plot without being reset to default values. Normally, the P1 and P2 positions which define the plot page on the plotter are set by the **plotlbe** command to their default locations. Use the **-noin** no-initialization option to override the default locations.

Examples

1. To send the file `longaxis.gl` to the `plt` plotter queue and specify to the backend that the file requires five frames to plot, enter:

   ```
   enq -Pplt -o -fr=5 longaxis.gl
   ```

2. To send the file `plotdata.gl` to the `plt` plotter queue, specifying that the plot page positions are not to be reset to default for this file, enter:

```
enq -Pplt -o -noin plotdata.gl
```

3. To send the file `twoplot.gl` to the `plt` plotter queue, specifying no plot page initialization and that the plotter print the drawing in two frames, enter:

```
enq -Pplt -o -noin -o fr=2 twoplot.gl
```

Files

/usr/lpd/plotIbe	Contains the **plotIbe** command.

Related Information

The **enq** command, **plotgbe** command.

Printer Overview for System Management in *AIX 5L Version 5.2 Guide to Printers and Printing* .

---

pmcycles Command

Purpose

Measures processor clock speed.

Syntax

pmcycles [**-d**] [**-m**]

Description

The **pmcycles** command uses the Performance Monitor cycle counter and the processor real-time clock to measure the actual processor clock speed in MHz. Optionally, it also displays the decrementer speed in MHz and nanoseconds per tick. The decrementer is a binary counter which generates a clock interrupt each time the clock goes to zero. The tick is the value of a decrement. On some machines, time is decremented in nanoseconds, so each tick is equal to one nanosecond. On other machines, the value of the decrement depends on the machine.

This command is only supported on processors supported by **bos.pmapi**.

Flags

-d	Displays the decrementer in MHz and nanoseconds per tick.
-m	Displays the speed of each of the processors.

Examples

1. To display the processor speed, type:

```
pmcycles
```

   Output similar to the following appears:

```
This machine runs at 133 MHz
```

2. To display each processor speed, type:

```
pmcycles -m
```

   Output similar to the following appears:

```
Cpu 0 runs at 200 MHz
CPU 1 runs at 200 MHz
```

Related Information

pm_cycles subroutine.

---

pop3d Daemon

Purpose

Starts the Post Office Protocol Version 3 (POP3) server process. This command only applies to AIX 4.2.1 or later.

Syntax

pop3d [**-c**]

Description

The **pop3d** command is a POP3 server. It supports the POP3 remote mail access protocol. Also, it accepts commands on its standard input and responds on its standard output. You normally invoke the **pop3d** command with the **inetd** daemon with those descriptors attached to a remote client connection.

The **pop3d** command works with the existing mail infrastructure consisting of **sendmail** and **bellmail.**

Flags

-c	Suppresses the reverse host name lookup.

Parameters

None

Exit Status

All error and status information is written to a logfile if **syslogd** is configured for logging.

Security

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

Auditing Events: N/A

Files

/usr/sbin/pop3d	Contains the **pop3d** command.
/etc/services	Specifies the file with port assignments for required services. The following entry must be in this file: `pop3 110/tcp postoffice3 # Post Office Protocol Ver. 3`

Related Information

Daemons: **imapd**

portmap Daemon

Purpose
Converts RPC program numbers into Internet port numbers.

Syntax
/usr/sbin/portmap

Description
The **portmap** daemon converts RPC program numbers into Internet port numbers.

When an RPC server starts up, it registers with the **portmap** daemon. The server tells the daemon which port number it is listening to and which RPC program numbers it serves. Thus, the **portmap** daemon knows the location of every registered port on the host and which programs are available on each of these ports.

A client consults the **portmap** daemon only once for each program the client tries to call. The **portmap** daemon tells the client which port to send the call to. The client stores this information for future reference.

Since standard RPC servers are normally started by the **inetd** daemon, the **portmap** daemon must be started before the **inetd** daemon is invoked.

> **Note:** If the **portmap** daemon is stopped or comes to an abnormal end, all RPC servers on the host must be restarted.

Flags
None

Examples
1. To start the **portmap** daemon, enter the following command:
   ```
   startsrc -s portmap
   ```
2. To stop the **portmap** daemon enter the following command:
   ```
   stopsrc -s portmap
   ```

Files

inetd.conf	Starts RPC daemons and other TCP/IP daemons.
/etc/rpc	Contains a list of server names and their corresponding **rpc** program numbers and aliases.

Related Information
The **rpcinfo** command.

The **inetd** daemon.

Network File System (NFS) Overview for System Management in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices*.

System Resource Controller Overview in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*.

TCP/IP Protocols in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices*.

portmir Command

Purpose

Allows one TTY stream (monitor) to attach to another TTY stream (target) and monitor the user session that is taking place on that stream.

Syntax

portmir { **-d** *mir_modem* **-t** *target* [**-m** *monitor*] | **-t** *target* [**-m** *monitor*] | { **-o** | **-c** *monitor* | **-q** }

Description

The **portmir** command allows one TTY stream (monitor) to attach to another TTY stream (target) and monitor the user session that is taking place on that stream. This is accomplished by pushing a special ″mirror″ module into both the target and monitor TTY streams.

Both the target and monitor TTYs receive a printed message on their respective displays when a monitoring session begins. The monitoring session can be terminated from either the target TTY, monitor TTY, or a third TTY not involved in the monitoring session. When the monitor is used in a non-service mode, both streams must be in the open state (that is, either a getty or active session must be taking place on each TTY) in order for the command to work. This is necessary to allow the pushing of the ″mirror″ streams module. The **portmir** command is supported for use with TTY devices only (PTS, TTY, LFT).

The terminal type, as defined in the TERM environment variable, must be the same for both the monitor and target TTY. The value of this environment variable must correspond to a valid entry in the **terminfo** database. An example terminal type would be ibm3151 or vt100. The LFT is similar to the vt100. Terminal emulators such as aixterm are usually similar in function to vt100.

Although the console can be used as either the target TTY or the monitor TTY, using the console as the monitor TTY is not recommended. However, if the console is used as the monitor TTY, note that the console is first automatically redirected to the target TTY for the duration of the monitoring session. When the monitoring session is terminated, the console is redirected back to the device specified in the CuAt ODM database attribute **syscons**. If the console had been previously redirected, the redirection is not preserved.

Async devices that provide offloading of character processing may have problems if they are mirroring devices that rely on the line discipline (**ldterm**) to provide this function. An example of this would be the 128-port async adapter. Use the **chdev** command to disable the fastcook attribute if a port of a dissimilar adapter is monitored. Run the command as follows:

```
chdev -l tty1 -a fastcook -disable
```

You can use the Devices application in Web-based System Manager (wsm) to change device characteristics.

Flags

-c *monitor*	Configures port for service boot by creating CuAt ODM database attribute **portmir_monitor**, which contains the **device** parameter as the value field. This device is used later as the default monitoring device when the **portmir** command is invoked in service mode (**-s** flag). Mirroring must be configured by the system administrator to execute at service boot time using the **-c** option. The target defaults to the device defined in the **portmir_monitor** attribute.
-d *mir_modem*	Sets monitoring port for dial-in purposes. Only the root user can issue the command with this flag. Ensure that **/usr/share/mir_modem** is linked to the correct modem setup file. **/usr/share/mir_modem** contains example files; you may need to create your own, depending on your type of modem.
-m *monitor*	Specifies monitoring device. If neither the **-m** option nor the **-s** option are specified, then the monitoring device defaults to the port on which the **portmir** command was run.
-o	Turns off monitoring and terminates the command.
-q	Queries the value set with the **-c** option.
-t *target*	Specifies target device to be monitored.

Security

Only a single mirror session may be running at any one time.

To mirror a port in the nonservice mode, place a list of users who may monitor them in a **.mir** file in your home directory (not required for the root user). When the **mirror** daemon begins running, the daemon checks to see who is on that port. It then checks to see if the user of the monitoring port is authorized to monitor that port. The **mirrord** command also validates the monitoring port's user.

The **.mir** file must have the format of a single user ID per line.

> **Attention:** Running the **su** command to change to root user during a mirror session gives root authority to *both* users.

Examples

1. After **user1** has placed **user2**'s login ID into **/u/user2/.mir** file, to mirror **user1** on target **tty1** from **user2** on monitor **tty2**, enter:
   ```
   portmir -t tty1 -m tty2
   ```
2. To mirror target **tty1** to user on monitor **tty2** who is dialing in, enter:
   ```
   portmir -t tty1 -m tty2 -d mir_modem
   ```
3. To set up mirroring for service boot, specifying the monitoring device during the service boot, enter:
   ```
   portmir -c tty
   ```
4. To disable mirroring during the service boot, enter:
   ```
   portmir -c off
   ```
5. To query the service boot mirroring device, enter:
   ```
   portmir -q
   ```

Files

/usr/share/modems/mir_modem	Modem configuration file examples for setting up dial-in.
/usr/sbin/portmir	Contains the command file.

Related Information

The **chdev** command.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.2 Web-based System Manager Administration Guide*.

---

post Command

Purpose

Routes a message.

Syntax

post [**-alias** *File ...*] [**-format** | **-noformat**] [**-msgid** | **-nomsgid**] [**-filter** *File* | **-nofilter**] [**-width** *Number*] [**-verbose** | **-noverbose**] [**-watch** | **-nowatch**] *File*

Description

The **post** command routes messages to the correct destinations. The **post** command cannot be started by the user. The **post** command can be called only by other programs.

The **post** command searches a message for all components that specify a recipient's address and parses each address to check for the proper format. The **post** command then puts addresses into the standard format and calls the **sendmail** command. The **post** command also performs header operations, such as appending the `Date:` and `From:` components and processing the `Bcc:` component. The **post** command uses the *File* parameter to specify the name of the file to be posted.

> **Note:** The **post** command may report errors when parsing complex addresses (for example, `@A:harold@B.UUCP`). If you use complex addresses, use the **spost** command instead of the **post** command.

Flags

-alias *File*	Searches the specified mail alias file for addresses. This flag may be repeated to specify multiple mail alias files. The **post** command automatically searches the **/etc/mh/MailAliases** file.
-filter *File*	Uses the header components in the specified file to copy messages sent to `Bcc:` recipients.
-format	Puts all recipient addresses into a standard format for the delivery transport system. This flag is the default.
-help	Lists the command syntax, available switches (toggles), and version information. **Note:** For Message Handler (MH), the name of this flag must be fully spelled out.
-msgid	Adds a message-identification component (such as `Message-ID:`) to the message.
-nofilter	Strips the `Bcc:` header from the message for the `To:` and `cc:` recipients. Sends the message with minimal headers to the `Bcc:` recipients. This flag is the default.
-noformat	Does not alter the format of the recipient addresses.
-nomsgid	Does not add a message-identification component to the message. This flag is the default.
-noverbose	Does not display information during the delivery of the message to the **sendmail** command. This flag is the default.
-nowatch	Does not display information during delivery by the **sendmail** command. This flag is the default.
-verbose	Displays information during the delivery of the message to the **sendmail** command. This information allows you to monitor the steps involved.

-watch Displays information during the delivery of the message by the **sendmail** command. This information allows you to monitor the steps involved.

-width *Number* Sets the width of components that contain addresses. The default is 72 columns.

Files

/etc/mh/MailAliases	Contains the default mail aliases.
/etc/mh/mtstailor	Contains MH command definitions.

Related Information

The **ali** command**, conflict** command, **mhmail** command, **send** command, **sendmail** command, **spost** command, **whom** command.

The **.mh_alias** File Format.

Mail Overview in *AIX 5L Version 5.2 System User's Guide: Communications and Networks*.

---

pppattachd Daemon

Purpose

Attaches an asynchronous device stream to the PPP (Point to Point Protocol) subsystem. Can be invoked as a daemon or a normal process.

Syntax

To Use a Specific tty Port as a Connection (Runs as a Daemon):

pppattachd /dev/tty*PortNumber* { **client** | **server** | **demand** } [**multilink**] [**connect** *"ConnectorProgram"*] [**inactive** *Seconds*] [**authenticate pap** | **chap**] [**peer pap** | **chap**] [**user** *Name*] [**remote** *HostName*] [**nodaemon**]

To Use Standard In and Standard Out as the tty Device (Runs as a Process):

pppattachd { **client** | **server** | **demand** } [**multilink**] [**inactive** *Seconds*] [**authenticate pap** | **chap**] [**peer pap** | **chap**] [**user** *Name*] [**remote** *HostName*] [**nodaemon**]

Description

The **pppattachd** daemon provides the mechanism to bind an asynchronous stream to the PPP subsystem. When placing an out going connection on a specific tty port, **pppattachd** becomes a daemon. When using stdin (standard in) and stdout (standard out) as the tty device for PPP communications **pppattachd** does not become a daemon. (It would be executed from the **$HOME/.profile** upon login on a tty device.)

You can activate PAP or CHAP authentication with the **authenticate** and **peer** options. Use the **smit** command to create entries in either the **/etc/ppp/pap-secrets** or **/etc/ppp/chap-secrets** file. The **pppattachd** daemon uses the passwords in these files to authenticate the connection. It searches only the **/etc/ppp/pap-secrets** file for PAP authentication and the **/etc/ppp/chap-secrets** file for CHAP authentication.

The multilink option is to used to identify the PPP link as having several attachments between the two PPP peers. PPP packets are fragmented at one peer, sent over the multiple attachments, and then reconnected on the remote peer that must also support multilink. The maximum receive reconstruction unit (MMRU) and endpoint descriptor are set through SMIT on the PPP Link Configuration menu. MRRU is the maximum data size before fragmentation. The endpoint discriminator uniquely identifies the local system.

Errors and messages are logged using the **syslog** facility.

Options

authenticate pap | **chap**	Defines the current system as the authenticator of either PAP or CHAP.
client | **server** | **demand**	Defines the type of subsystem connection to be bound to on the system running the daemon.
connect *"ConnectorProgram"*	Specifies the program to use to place an outgoing connection. The tty device opened is passed as stdin and stdout to the program. The **/usr/sbin/pppdial** command is a connector program that can be used.
inactive *Seconds*	Specifies the number (unsigned integer) of seconds to wait for inactivity on the link before terminating the connection. The default value is 0 (no timeout).
multilink	Identifies the PPP link as having a group of attachments connecting to two PPP peers.
nodaemon	Specifies to the attachment process that it is not to become a daemon. You must use this option for attachment processes that are invoked with demand connections.
peer pap | **chap**	Defines the current system as the peer of either PAP or CHAP.
remote *HostName*	Defines the remote host name to be used for PAP authentication. An entry for *UserName RemoteHostName Password* must exist in **/etc/ppp/pap-secrets** file for a successful connection. This option only has meaning for PAP authentication on both the authenticator and peer.
user *Name*	Defines the user entry to use for PAP authentication. An entry for *UserName RemoteHostName Password* must exist in **/etc/ppp/pap-secrets** file for a successful connection. This option only has meaning for PAP authentication on the peer.

Exit Status

This command returns the following exit values:

0	Successful completion.
!0	An error occurred.

Security

Access Control: Any User

Auditing Events: N/A

Examples

1. You want System A to act as a client to server System B. From System A enter:

   ```
   /usr/sbin/pppattachd /dev/tty0 client connect "sysbconnector"
   ```

   where `sysbconnector` is the connector program.

   On System B, the user that logged in would have invoked from **$HOME/.profile**:

   ```
   exec /usr/sbin/pppattachd server 2>/dev/null
   ```

2. You want server System B to contact client System A. From System B enter:

   ```
   /usr/sbin/pppattachd /dev/tty0 server connect "sysaconnector"
   ```

   where `sysaconnector` is the connector program.

   On System A, the user that logged in would have invoked from **$HOME/.profile**:

   ```
   exec /usr/sbin/pppattachd client 2>/dev/null
   ```

3. You want System A to act as a client to server System B using PAP authentication. System B acts as the authenticator and System A is the peer to be authenticated. From System A enter:

   ```
   /usr/sbin/pppattachd /dev/tty0 client peer pap user username \
   connect "sysbconnector"
   ```

   where `sysbconnector` is the connector program.

   On System A, the **/etc/ppp/pap-secrets** file contains: username * ppppassword. On System B, the user that logged in would have invoked from **$HOME/.profile**:

   ```
   exec /usr/sbin/pppattachd server authenticate pap 2>/dev/null
   ```

   On System B, the **/etc/ppp/pap-secrets** file contains: username * ppppassword.

Files

/usr/sbin/pppattachd	Contains the **pppattachd** daemon.
/etc/ppp/att*XXX*.pid	Contains the process id. *XXX* is the pid, the content of the file is the network layer ID to which the attachment was bound. The user must belong to uucp group for the pid file to be created.

Related Information

The **pppcontrold** daemon, **pppdial** command.

The **syslog** subroutine.

The **profile** file format.

Asynchronous Point-to-Point Protocol (PPP) Subsystem Overview in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

---

pppcontrold Daemon

Purpose

Controls startup and management of the PPP (Point to Point Protocol) subsystem.

Syntax

To Start and Stop by Using the System Resource Controller:

startsrc -s pppcontrold

stopsrc -s pppcontrold

Description

The **pppcontrold** daemon reads in the **/etc/ppp/lcp_config** and **/etc/ppp/if_conf** files to install and configure the PPP subsystem. SMIT should be used to generate both /etc/ppp/lcp_config and /etc/ppp/if_conf. To modify these files the user must have root authority or be a member of the uucp group. The configuration files are read at initialization where the appropriate streams modules are configured and loaded, and the tcpip network interface layers are installed into the system. After configuring the subsystem, the **pppcontrold** daemon monitors the streams associated with the IP interfaces to perform operations such as setting IP addresses, and the flags of the IP interface. The **pppcontrold** daemon terminates upon reciept of SIGTERM or when the **stopsrc** command is invoked. The prefered method of starting and stopping the **pppcontrold** daemon is with SRC (System Resource Controller). You must have root authority to run the src commands.

Errors and messages are logged using the **syslog** facility.

The **pppcontrold** daemon creates the **/etc/ppp/pppcontrold.pid** file, which contains a single line with the command process ID used to terminate the **pppcontrold** daemon.

Flags

None

/etc/ppp/lcp_config File

This file provides the configuration information required for the subsystem. These values are used to ensure proper allocation of storage at the time the subsystem is configured. It is important to configure just what is needed since these values define storage that is allocated within the kernel. Blank lines and lines beginning with a # (pound sign) are ignored in the configuration file. Do not use blank lines or lines beginning with # (pound sign) within the interface definition. Only use these lines between interface definitions.

Required Keywords

server_name *name*	Name of this system. This name should be unique to the system. Ensure that the first 20 bytes of the name are unique.
lcp_server #	Number of server connections. Represents the number of server connections that the subsystem will allow.Storage for all specified connections is allocated at the time the subsystem is configured. The minimun value is 0 and the maximum value is gated by the amount of memory in the system.
lcp_client #	Number of client connections. The minimum value is 0 and the maximum value is gated by the amount of memory in the system. Client connections are IP interfaces configured without addresses.
num_if #	Number of IP interfaces to configure,. Must be less than or equal to lcp_server + lcp_client.
num_hdlc #	Number of async hdlc modules to configure for. This will define the maxmimum number of concurent asynchronous PPP sessions (client and server) that can be active. Cannot be greater than lcp_server lcp_client.

Optional Keywords

These keywords will override the global default LCP options.

txacm *0xXXXXXXXX*	Transmit Asynchronous Character Map.
-negacm	Do not negotiate async character mapping. Rejects the peers configuration information frames that contains this option.
-negmru	Do not negotiate MRU (Maximum Receive Unit). Rejects the peers configuration information frames that contains this option.
mru #	MRU desired. A default is 1500.
-negacf	Do not negotiate ACF (address control field) compression. ACF will not be compressed. Rejects the peers configuration information frames that contain this option.
-negprotocolcompress	Do not negotiate protocol compression. Normaly the PPP protocol field will be compressed by one byte for Network protocols. This disables negotiation of this option for both receiveing and sending frames.

/etc/ppp/if_conf File

This file defines all the server TCP/IP interfaces. Blank lines and lines beginning with a # (pound sign) are ignored in the configuration file. Do not use blank lines or lines beginning with # (pound sign) within the interface definition . Only use these lines between interface definitions.

Keywords

interface Indicates that a new interface defintion is being started.

server Indicates that the interface is a server connection.

Requires the following keywords:

local_ip xxx.yyy.zzz.qqq

remote_ip xxx.yyy.zzz.qqq

These addresses MUST be different on the pair basis, however the local IP address can be the same for all PPP interfaces. On a given server, the remote address must be unique.

Optional Keywords

netmask xxx.xxx.xxx.xxx Specifies a netmask for this interface.

Exit Status

This command returns the following exit values:

0 Successful completion.

!0 An error occurred.

Security

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

Examples

Example /ect/ppp/lcp_config File:

```
# Comment line
server_name PPPSERVER_ALPHA
lcp_server 5
lcp_client 1
num_if 6
num_hdlc 6
```

Example /ect/ppp/if_conf File:

```
# Sample ip server configuration information.
# Note that the complete stanza does not contain
# comments or blank lines
interface
server
local_ip 129.35.130.45
remote_ip 129.35.131.191
netmask 255.255.240.0

#However between stanzas one can have blank or
# comment lines.

interface
server
local_ip 129.35.130.45
remote_ip 129.35.131.196
netmask 255.255.240.0
```

```
interface
server
local_ip 129.35.130.45
remote_ip 129.35.131.197
netmask 255.255.240.0

interface
server
local_ip 129.35.130.45
remote_ip 129.35.131.201
netmask 255.255.240.0

interface
server
local_ip 129.35.130.45
remote_ip 129.35.131.212
netmask 255.255.240.0
```

The above configuration files would result in a subsystem that installs the IP interfaces as follows:

```
pp0: flags=71<UP,POINTOPOINT,NOTRAILERS>
     inet 129.35.130.45 --> 129.35.131.191 netmask 0xfffff000
pp1: flags=31<UP,POINTOPOINT,NOTRAILERS>
     inet 129.35.130.45 --> 129.35.131.196 netmask 0xfffff000
pp2: flags=31<UP,POINTOPOINT,NOTRAILERS>
     inet 129.35.130.45 --> 129.35.131.197 netmask 0xfffff000
pp3: flags=31<UP,POINTOPOINT,NOTRAILERS>
     inet 129.35.130.45 --> 129.35.131.201 netmask 0xfffff000
pp4: flags=31<UP,POINTOPOINT,NOTRAILERS>
     inet 129.35.130.45 --> 129.35.131.212 netmask 0xfffff000
pp5: flags=30<POINTOPOINT,NOTRAILERS>
     inet 0.0.0.0 --> 0.0.0.0 netmask 0xff000000
```

Note: pp5 is the result of the lcp_client keyword in the /etc/ppp/lcp_config file (lcp_client 1).

Files

/usr/sbin/pppcontrold	Contains the **pppcontrold** daemon.
/etc/ppp/lcp_config	Configures the subsystem (**lcp_config** should be generated by SMIT).
/etc/ppp/if_conf	Configures the TCP/IP interfaces (**if_conf** should be generated by SMIT).
/etc/ppp/pppcontrold.pid	Contains the **pppcontrold** process id.
/etc/ppp/ppp.conf	Contains input to the **strload** command.

Related Information

The **pppattachd** daemon, **pppdial** command.

The **startsrc** command, **stopsrc** command.

The **syslog** subroutine.

Asynchronous Point-to-Point Protocol (PPP) Subsystem Overview in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices*.

The System Resource Controller Overview in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices* gives an explanation of subsystems, subservers, and the System Resource Controller.

pppdial Command

Purpose

Establish an asynchronous connection with a remote system for use by the PPP (Point to Point Protocol) subsystem.

Syntax

pppdial [**-t** *TimeOut*] [**-v**] [**-d** *VerboseFile*] **-f** *ChatFile*

Description

The **pppdial** command provides the capability to establish a connection with a remote system over an asynchronous device. It is used with the **pppattachd** daemon as the means for carrying out the dialog with modems and remote systems to the point where PPP frames should be sent. The **pppdial** command uses standard input (stdin) and standard output (stdout) as the devices over which the dialog occurs.

Errors and messages are logged using the **syslog** facility.

Flags

-d *VerboseFile*	Logs the chat activity to *VerboseFile*. If *VerboseFile* does not exist, the **pppdial** command creates it. If *VerboseFile* does exists, the **pppdial** command appends the output to the existing file.
-f *ChatFile*	Specifies the file which contains the dialog that is to occur over the tty device. The content of *ChatFile* conforms to the syntax of the Basic Networking Utility (BNU)/UNIX to UNIX Copy Program (UUCP).
-t *TimeOut*	Specifies the number of seconds to wait before timing out during the **Expect** phase of the chat activity.
-v	Logs the chat activity using the syslog facility.

Exit Status

This command returns the following exit values:

0	Successful completion.
!0	An error occurred.

Security

Access Control: Any User

Examples

To establish a connection with a remote system, enter on the command line in one line:

```
/usr/sbin/pppattachd client /dev/tty0 connect "/usr/sbin/pppdial
-v -f /home/pppuser/dialer.file"
```

The *ChatFile* named `/home/pppuser/dialer.file` contains:

```
''
atdt4311088
CONNECT
\\d\\n
ogin
pppuser
ssword
pppuserpwd
```

with the following meaning:

```
''            Expect a nul string
atdt4311088   Send the modem the dial command
              4311088 is the phone number to dial
CONNECT       Expect connect from the modem
\\d\\n        Delay for 1 second then send a new line
ogin          Expect the string ogin
pppuser       Send the string pppuser
              pppuser is the user id on the remote system
ssword        Expect the string ssword
pppuserpwd    Send the string pppuserpwd
              pppuserpwd is the password of the user pppuser on the
              remote system
```

The remote system must have a user `pppuser` defined with a password `pppuserpwd` and a **$HOME/.profile** containing:

```
exec pppattachd server 2>/dev/null
```

This is a very simplistic example. The example requires that the PPP subsystem is running on both the client and server (or remote) system. The example requires that the client system have a modem defined on `/dev/tty0`. The *ChatFile* contains the number 4311088 to dial. The remote system must also have a user defined with a password and a **.profile** which starts a PPP attachment on the remote system. The device (`/dev/tty0`), phone number, user, user password and mechanism starting the PPP attachment are variable and should reflect the current values on the server system.

Files

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

Related Information

The **pppattachd** daemon, **pppcontrold** daemon.

The **syslog** subroutine.

Asynchronous Point-to-Point Protocol (PPP) Subsystem Overview in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

---

pppstat Command

Purpose

Extracts and displays the RAS (Reliability, Availability, and Serviceability) information of the PPP (Point to Point Protocol) subsystem. This command only applies to AIX 4.1.5.

Syntax

pppstat

Description

The **pppstat** command provides the capability to monitor particular characteristics of active links. The following information is displayed for all active links:

LCP Multiplexing Layer

Local MRU	Specifies the Maximum Receive Unit setting for this host. This is maximum length of a packet that the remote host can send to the local host.
Remote MRU	Specifies the Maximum Receive Unit setting for the remote host. This is the maximum length of a packet that we can send to the remote host.
Local To Peer ACCM	Specifies the ASYNC Character Map used in the transmission of packets to the remote host.
Peer To Local ACCM	Specifies the ASYNC Character Map used by the remote host in the transmission of packets to the local host.
Local To Remote Protocol Field Compression	Specifies whether Protocol Compression is used in the transmission of packets to the remote host.
Remote To Local Protocol Field Compression	Specifies whether Protocol Compression is used in the transmission of packets from the remote host to the local host.
Local To Remote Address/Control Field Compression	Specifies whether Address/Control field compression is being used in the transmission of packets to the remote host.
Remote To Local Address/Control Field Compression	Specifies whether Address/Control field compression is being used in the transmission of packets from the remote host to the local host.

LCP Multiplexing Layer prior to PPP negotiating

MRU	Specifies the Maximum Receive Unit for receiving packets. This is the value that this host attempted to negotiate with the remote host.
Receive ACCM	Specifies the initial remote-to-local ASYNC Character Map that was used in the negotiation.
Transmit ACCM	Specifies the initial local-to-remote ASYNC Character Map that was used in the negotiation.
Magic Number	Specifies the magic number attempted in negotiation.
Frame Check Size	Specifies the length of the Frame Check Sequence that this host attempted to negotiate. This is fixed at 16 bits.

HDLC Framing Layer

Bad Address Fields	Specifies the number of times a packet has been received with an incorrect address field.
Bad Controls Fields	Specifies the number of times a packet has been received with an incorrect control field.
Oversized Packets	Specifies the number of times a packet has been received that has a length that exceeds the Maximum Receive Unit length.
Bad Frame Check Sequence	Specifies the number of times a packet has been received with a bad Frame Check Sequence.
Incoming Good Octets	Specifies the number of octets received in valid packets.
Outgoing Good Octets	Specifies the number of octets sent successfully in packets.
Incoming Good Packets	Specifies the number of packets received successfully.
Outgoing Good Packets	Specifies the number of packets sent successfully.

The output is sent to **stdout**. Messages are sent to **stderr**.

Exit Status

This command returns the following exit values:

0 Successful completion.
>0 An error occurred.

Security

Access Control: Any User

Auditing Events: N/A

Files

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

Related Information

Commands: **pppdial**.

Daemons: **pppcontrold**, **pppattachd**.

Files: **profile**.

Subroutines: **syslog**

Asynchronous Point-to-Point Protocol (PPP) Subsystem Overview in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

---

pprof Command

Purpose

Reports CPU usage of all kernel threads over a period of time.

Syntax

pprof { *time* | **-l** *pprof.flow* | **-i** *tracefile* | **-d** } [**-T** *bytes*] [**-v**] [**-s**] [**-n**] [**-f**] [**-p**] [**-w**]

Description

The **pprof** command reports on all kernel threads running within an interval using the **trace** utility. The raw process information is saved to **pprof.flow**, and five reports are generated. The **pprof** command can also take previously generated **Pprof.flow** to regenerate reports. If no flags are specified, all reports are generated.

Types of Reports

pprof.cpu	Lists all kernel level threads sorted by actual cpu time. Contains: Process Name, Process ID, Parent Process ID, Process State at Beginning and End, Thread ID, Parent Thread ID, Actual CPU Time, Start Time, Stop Time, Stop - Start
pprof.start	Lists all kernel threads sorted by start time. Contains: Process Name, Process ID, Parent Process ID, Process State Beginning and End, Thread ID, Parent Thread ID, Actual CPU Time, Start Time, Stop Time, Stop - Start

pprof.namecpu	Lists information about each type of kernel thread (all executable with the same name). Contains: Process Name, Number of Threads, CPU Time, % of Total CPU Time
pprof.famind	Lists all processes grouped by families (processes with a common ancestor). Child process names are indented with respect to the parent. Contains: Start Time, Stop Time, Actual CPU Time, Process ID, Parent Process ID, Thread ID, Parent Thread ID, Process State at Beginning and End, Level, Process Name.
pprof.famcpu	Lists the information for all families (processes with a common ancestor). The Process Name and Process ID for the family is not necessarily the ancestor. Contains: Start Time, Process Name, Process ID, Number of Threads, Total CPU Time.

Flags

-d	Waits for the user to execute **trcon** and **trcstop** from the command line.
-f	Specifies to only generate the **pprof.famcpu** and **pprof.famind** reports.
-l *pprof.flow*	Indicates to generate reports from a previously generated **pprof.flow**. Specifies to only generate the **pprof.namecpu** report.
-i *tracefile*	Indicates to generate reports from a **tracefile**. The trace must contain the following hooks: 135,106,10C,134,139,465,467,00A
-n	Specifies to only generate the **pprof.namecpu** report.
-p	Specifies to only generate the **pprof.cpu** report.
-s	Specifies to only generate the **pprof.start** report.
-T	Sets the trace kernel buffer size in bytes. The default is 32000.
-v	Sets verbose mode (print extra details).
-w	Specifies to only generate **pprof.flow**.
time	Specifies the number of seconds to trace the system.

Note: Review the **/usr/lpp/perfagent/README.perfagent.tools** file for the latest on changes to the performance analysis tools.

Related Information

The **trace** command, **trcrpt** command, **filemon** command.

The **trcon** subroutine, **trcstop** subroutine.

---

pr Command

Purpose

Writes a file to standard output.

Syntax

pr [**+***Page*] [**-***Column* [**-a**] | **-m**] [**-d**] [**-F**] [**-r**] [**-t**] [**-e** [*Character*] [*Gap*]] [**-h** *Header*] [**-i** [*Character*] [*Gap*]] [**-l** *Lines*] [**-n** [*Character*] [*Width*]] [**-o** *Offset*] [**-s** [*Character*]] [**-w** *Width*] [**-x** [*Character*] [*Width*]] [**-f**] [**-p**] [*File* ... | **-**]

Description

The **pr** command writes the specified file or files to standard output. If you specify the **-** (minus sign) parameter instead of the *File* parameter, or if you specify neither, the **pr** command reads standard input. A heading that contains the page number, date, time, and name of the file separates the output into pages.

Unless specified, columns are of equal width and separated by at least one space. Lines that are too long for the page width are cut off. If standard output is a workstation, the **pr** command does not display error messages until it has ended.

Flags

-*Column*	Sets the number of columns to the value specified by the *Column* variable. The default value is 1. This option should not be used with the **-m** flag. The **-e** and **-i** flags are assumed for multicolumn output. A text column should never exceed the length of the page (see the **-l** flag). When the **-***Column* flag is used with the **-t** flag, use the minimum number of lines to write the output.
+*Page*	Begins the display with the page number specified by the *Page* variable. The default value is 1.
-a	Modifies the effect of the **-***Column* flag so that multiple columns are filled horizontally, from left to right. For example, if there are two columns, the first input line goes in column 1, the second goes in column 2, the third becomes line 2 of column 1, and so forth. If the **-a** flag is not specified, columns are created vertically.
-d	Produces double-spaced output.
-e[*Character*][*Gap*]	Expands tabs to character positions as follows: *Gap*+1, 2**Gap*+1, 3**Gap*+1, and so on. The default value of *Gap* is 8. Tab characters in the input expand to the appropriate number of spaces in order to line up with the next tab setting. If you specify a value for the *Character* variable (any character other than a digit), that character becomes the input tab character. The default value of the *Character* variable is the ASCII TAB character.
-F	Uses a form-feed character to advance to a new page. (Otherwise the **pr** command issues a sequence of line-feed characters.) Pauses before beginning the first page if the standard output is a workstation. This flag is equivalent to the **-f** flag.
-f	Uses a form-feed character to advance to a new page. (Otherwise the **pr** command issues a sequence of line-feed characters.) Pauses before beginning the first page if the standard output is a workstation. This flag is equivalent to the **-F** flag.
-h *Header*	Uses the specified header string as the page header. If the **-h** flag is not used, the page header defaults to the file name specified by the *File* parameter.
-i[*Character*][*Gap*]	Replaces white space wherever possible by inserting tabs to character positions, as follows: *Gap*+1, 2**Gap*+1, and 3**Gap*+1, and so forth. The default value of *Gap* is 8. If you specify a value for the *Character* variable (any character other than a digit), that character is used as the output tab character.
-l *Lines*	Overrides the 66-line default and resets the page length to the number of lines specified by the *Lines* variable. If the *Lines* value is smaller than the sum of both the header and trailer depths (in lines), the header and trailer are suppressed (as if the **-t** flag were in effect).
-m	Merges files. Standard output is formatted so the **pr** command writes one line from each file specified by the *File* parameter, side by side into text columns of equal fixed widths, based on the number of column positions. This flag should not be used with the **-** *Column* flag.
-n[*Character*][*Width*]	Provides line numbering based on the number of digits specified by the *Width* variable. The default is 5 digits. The line number occupies the first *Width*+1 column positions of each text column of default output, or of each line of output when the **-m** flag is set. If the *Character* variable is specified (any non-digit character), it is appended to the line number to separate it from what follows on the line. The default character separator is the tab character.
-o *Offset*	Indents each line by the number of character positions specified by the *Offset* variable. The total number of character positions per line is the sum of the width and offset. The default *Offset* value is 0.
-p	Pauses before beginning each page if the output is directed to a workstation. The **pr** command sounds the alarm at the workstation and waits for you to press the Enter key.

-r	Does not display diagnostic messages if the system cannot open files.
-s[*Character*]	Separates columns by the single character specified by the *Character* variable instead of by the appropriate number of spaces. The default value for the *Character* variable is an ASCII TAB character.
-t	Does not display the five-line identifying header and the five-line footer. Stops after the last line of each file without spacing to the end of the page.
-w *Width*	Sets the width of line to width column positions for multiple text-column output only. If the **-w** option is not specified and the **-s** option is not specified, the default width is 72. If the **-w** is not specified and the **-s** option is specified, the default width is 512. For single column output, input lines will not be truncated.
-x[*Character*][*Width*]	Provides the same line numbering functions as the **-n** flag.

Exit Status

This command returns the following exit values:

0	All files were successfully written.
>0	An error occurred.

Examples

1. To print a file with headings and page numbers on the printer, type:

   ```
   pr prog.c | qprt
   ```

   This adds page headings to the **prog.c** file and sends it to the **qprt** command. The heading consists of the date the file was last modified, the file name, and the page number.
2. To specify a title, type:

   ```
   pr  -h "MAIN PROGRAM" prog.c | qprt
   ```

   This prints the **prog.c** file with the title Main Program in place of the file name. The modification date and page number are still printed.
3. To print a file in multiple columns, type:

   ```
   pr -3 word.lst | qprt
   ```

   This prints the **word.lst** file in three vertical columns.
4. To print several files side by side on the paper:

   ```
   pr -m -h "Members and Visitors" member.lst visitor.lst | qprt
   ```

   This prints the **member.lst** and **visitor.lst** files side by side with the title Members and Visitors.
5. To modify a file for later use, type:

   ```
   pr -t -e prog.c > prog.notab.c
   ```

   This replaces tab characters in the **prog.c** file with spaces and puts the result in **prog.notab.c** file. Tab positions are at every eighth column (that is 9, 17, 25, 33, . . .). The **-e** flag tells the **pr** command to replace the tab characters; the **-t** flag suppresses the page headings.

Files

/usr/bin/pr	Contains the **pr** command.
/dev/tty*	Suspends messages.

Related Information

The **cat** command, **qprt** command.

Files Overview in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices* describes files, file types, and how to name files.

Input and Output Redirection Overview in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices* describes how the operating system processes input and output.

National Language Support Overview in *AIX 5L Version 5.2 National Language Support Guide and Reference* explains collating sequences, equivalence classes, and locale.

Shells Overview in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices* describes shells, the different types, and how they affect the way commands are interpreted.

File and Directory Access Modes in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices* introduces file ownership and permissions to access files and directories.

---

prctmp Command

Purpose

Displays the session record files.

Syntax

/usr/sbin/acct/prctmp *File*...

Description

A user with administrative authority can enter the **prctmp** command to display the session record file created by the **acctcon1** command, normally the **/var/adm/acct/nite/ctmp** file. The session record file is converted into the connect-time total accounting record by the **acctcon2** command and then incorporated into the daily accounting report.

Security

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

Example

To display the session record file, enter:

```
prctmp /var/adm/acct/nite/ctmp
```

This command displays the session record file created by the **acctcon1** command.

Files

/usr/sbin/acct	The path to the accounting commands.
/var/adm/acct/nite	Contains accounting data files.

Related Information

Accounting Overview in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*.

Setting Up an Accounting System in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*.

prdaily Command

Purpose
Creates an ASCII report of the previous day's accounting data.

Syntax
/usr/sbin/acct/prdaily [**-l**] [*mmdd*] [**-c**]

Description
The **prdaily** command is called by the **runacct** command to format an ASCII report of the previous day's accounting data. The report resides in the **/var/adm/acct/sum/rprt***mmdd* file, where *mmdd* specifies the month and day of the report.

Flags

-c	Reports exceptional resource usage by command. This flag may be used only on the current day's accounting data.
-l [*mmdd*]	Reports exceptional usage by login ID for the specified date. Use the *mmdd* variable to specify a date other than the current day.

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

Files

/usr/sbin/acct	The path to the accounting commands.
/usr/sbin/acct/ptelus.awk	Calculates the limits for exceptional usage by login ID. This is a shell procedure.
/usr/sbin/acct/ptecms.awk	Calculates the limits of exceptional usage by command name. This is a shell procedure.
/var/adm/acct/sum	Cumulative directory for daily accounting records.

Related Information
The **acctcms** command, **acctcom** command, **acctmerg** command, **runacct** command.

For more information about the Accounting System, the preparation of daily and monthly reports, and the accounting files, see the Accounting Overview in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*.

Setting Up an Accounting System in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices* explains the steps you must take to establish an accounting system.

preprpnode Command

Purpose
Prepares a node to be defined to a peer domain.

Syntax

preprpnode [**–k**] [**–h**] [**–TV**] *node_name1* [*node_name2* ...]

preprpnode [**–k**] [**–h**] [**–TV**] **–f** *file_name*

Description

The **preprpnode** command prepares security on the node on which the command is run so it can be defined in a peer domain. It allows for peer domain operations to be performed on this node and must be run before the node can join a peer domain using the **mkrpdomain** or **addrpnode** command.

Before the **mkrpdomain** command is issued on a node, the **preprpnode** command must be run on each node to be defined to the new peer domain, using the name of the node that is to run the **mkrpdomain** command as the parameter. This gives the **mkrpdomain** node the necessary authority to create the peer domain configuration on each new node and set up additional security.

Before the **addrpnode** command is issued on a node, the **preprpnode** command must be run on each node that is to be added, using the names of all online nodes as the parameters. This gives the online nodes the authority to perform the necessary operations on the new node.

The **preprpnode** command performs the following:

1. Establishes trust with the node names specified on the command by adding their public keys to the trusted host list.
2. Modifies the resource monitoring and control (RMC) access control list (ACL) file to enable access to peer domain resources on this node from the other nodes in the peer domain. This allows peer domain operations to occur on the node. The RMC subsystem is refreshed so that these access changes will take effect.
3. RMC remote connections are enabled.

If the nodes that are to be defined to a peer domain are already in a management domain, you do not need to exchange public keys. You can use the **-k** flag to omit this step.

Flags

–f *file_name*
: Reads a list of node names from *file_name*.

–k
: Specifies that the command should not exchange public keys.

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

–T
: Writes the command's trace messages to standard error. For your software service organization's use only.

-V
: Writes the command's verbose messages to standard output.

Parameters

node_name1 **[***node_name2* **...]**
: Specifies the node (or nodes) from which peer domain commands can be accepted. Typically, this is the name of the node that will be running the **mkrpdomain** command when forming the peer domain. When adding to the peer domain, it is a list of the nodes that are currently online in the peer domain. The node name is the IP address or the long or short version of the DNS host name. The node name must resolve to an IP address.

Security

The user of the **preprpnode** command needs write permission to the access control list (ACL) file. Permissions are specified in the ACL file. See *RSCT Guide and Reference* 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.

Restrictions

This command must run on a node that will be defined to the peer domain.

Standard Input

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

Standard Output

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

Standard Error

All trace messages are written to standard error.

Examples

1. Suppose **mkrpdomain** will be issued from **nodeA**. To prepare **nodeB**, **nodeC**, and **nodeD** to be defined to a new peer domain, **AppIDomain**, run this command on **nodeB**, on **nodeC**, and then on **nodeD**:

   ```
   preprpnode nodeA
   ```

2. Suppose **nodeA** and **nodeB** are online in **AppIDomain**. To prepare **nodeC** to be added to the existing domain, run this command on **nodeC**:

   ```
   preprpnode nodeA nodeB
   ```

   Alternatively, create a file called **onlineNodes** with these contents:

   ```
   nodeA
   nodeB
   ```

Then, run this command on **nodeC**:

```
preprpnode -f onlineNodes
```

Location

/usr/sbin/rsct/bin/preprpnode
Contains the **preprpnode** command

Files

The access control list (ACL) file **/var/ct/cfg/ctrmc.acls** is modified. If this file does not exist, it is created.

Related Information

Commands: **addrpnode**, **lsrpdomain**, **lsrpnode**, **mkrpdomain**

---

prev Command

Purpose

Shows the previous message.

Syntax

prev [**+***Folder*] [**-header** | **-noheader**] [**-showproc** *CommandString* | **-noshowproc**]

Description

The **prev** command displays the previous message in a folder. The **prev** command is similar to the **show** command with the **prev** value specified.

The **prev** command passes any flags that it does not recognize to the **showproc** program.

Flags

+*Folder*	Specifies the folder that contains the message you want to show.
-header	Displays a one-line description of the message being shown. The description includes the folder name and the message number. This flag is the default.
-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.
-noheader	Prevents display of a one-line description of each message.
-noshowproc	Uses the **/usr/bin/cat** command to list the previous command.
-showproc *CommandString*	Uses the specified command string to perform the listing.

Profile Entries

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

`Current-Folder:`	Sets the default current folder.
`Path:`	Specifies the *UserMhDirectory*.
`showproc:`	Specifies the program used to show messages.

Examples

1. To display the previous message in the current folder, enter:

   ```
   prev
   ```

   The system responds with a message similar to the following:

   ```
   (Message schedule: 10)
   ```

   The text of the message is also displayed. In this example, message `10` in the current folder `schedule` is the previous message.
2. To show the previous message in the `meetings` folder, enter:

   ```
   prev  +meetings
   ```

   The system responds with a message similar to the following:

   ```
   (Message inbox: 5)
   ```

   In this example, message 5 in the `meetings` folder is the previous message.

Files

$HOME/.mh_profile	Contains the MH user profile.
/usr/bin/prev	Contains the **prev** command.

Related Information

The **next** command, **show** command.

The **.mh_alias** file format, **.mh_profile** file format.

Mail Overview in *AIX 5L Version 5.2 System User's Guide: Communications and Networks*.

---

printenv Command

Purpose

Displays the values of environment variables.

Syntax

printenv [*Name*]

Description

The **printenv** command displays the values of environment variables. If you specify the *Name* parameter, the system only prints the value associated with the *Name* parameter. If you do not specify the *Name* parameter, the **printenv** command displays the current environment, showing one *Name* **=***Value* sequence per line.

If you specify a *Name* parameter that you have not defined in the environment, the **printenv** command returns an exit status of 1; otherwise it returns a status of 0 (zero).

Examples

1. To find the current setting of the **MAILMSG** environment variable, enter:

   ```
   printenv MAILMSG
   ```

2. The command returns the value of the **MAILMSG** environment variable. For example:
   ```
   YOU HAVE NEW MAIL
   ```

Related Information

The **env** command.

The **environment** file.

Profiles Overview in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*.

Shells Overview in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices*.

---

printf Command

Purpose

Writes formatted output.

Syntax

printf *Format* [*Argument* ...]

Description

The **printf** command converts, formats, and writes its *Argument* parameters to standard output. The *Argument* parameters are formatted under control of the *Format* parameter. The formatted output line cannot exceed **LINE_MAX** bytes in length.

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

LANG	Determines the locale to use for the locale categories when both **LC_ALL** and the corresponding environment variable (beginning with **LC_**) do not specify a locale.
LC_ALL	Determines the locale to be used to override any values for locale categories specified by the setting of **LANG** or any other **LC_** environment variable.
LC_CTYPE	Determines the locale for the interpretation of sequences of bytes of text data as characters; for example, single versus multibyte characters in parameters.
LC_MESSAGES	Determines the language in which messages should be written.
LC_NUMERIC	Determines the locale for numeric formatting. This environment variable affects the format of numbers written using the **e, E, f, g,** and **G** conversion characters.

The *Format* parameter is a character string that contains three types of objects:

- Plain characters copied to the output stream.
- Conversion specifications, each of which cause 0 or more items to be retrieved from the value parameter list.
- The following escape sequences. When copied to the output stream, these sequences cause their associated action to be displayed on devices capable of the action:

****	Backslash
\a	Alert
\b	Backspace
\f	Form feed
\n	New line
\r	Carriage return
\t	Tab
\v	Vertical tab

*****ddd* Where *ddd* is a one-, two-, or three-digit octal number. These escape sequences are displayed as a byte with the numeric value specified by the octal number.

The *Argument* parameter is a list of one or more strings to be written to standard output under the control of the *Format* parameter.

The *Format* parameter is reused as often as necessary to satisfy the *Argument* parameters. Any extra **c** or **s** conversion specifications are evaluated as if a null string *Argument* were supplied; other extra conversion specifications are evaluated as if a 0 *Argument* were supplied. Where the *Format* parameter contains no conversion specifications and *Argument* parameters are present, the results are unspecified.

Each conversion specification in the *Format* parameter has the following syntax in this order:

1. A **%** (percent sign).
2. Zero or more options, which modify the meaning of the conversion specification. The option characters and their meanings are:

- The result of the conversion is left-aligned within the field.

+ The result of a signed conversion always begins with a sign (+ or -).

blank If the first character of a signed conversion is not a sign, a blank is prefixed to the result. If both the blank and + option characters are displayed, then the blank option character is ignored.

This option specifies that the value is to be converted to an alternate form. For **c**, **d**, **i**, **u**, and **s** conversions, the option has no effect. For **o** conversion, it increases the precision to force the first digit of the result to be a, 0 (zero). For **x** and **X** conversions, a nonzero result has 0x, or 0X prefixed to it, respectively. For **e**, **E**, **f**, **g**, and **G** conversions, the result always contains a radix character, even if no digits follow the radix character. For **g** and **G** conversions, trailing zeros are not removed from the result as they usually are.

0 For **d**, **i**, **o**, **u**, **x**, **e**, **E**, **f**, **g**, and **G** conversions, leading zeroes (following any indication of sign or base) are used to pad to the field width, no space padding is performed. If the **0** (zero) and the **-** (minus sign) options are displayed, the **0** (zero) option is ignored. For **d**, **i**, **o**, **u**, **x**, and **X** conversions, if a precision is specified, the **0** (zero) option is ignored.

Note: For other conversions, the behavior is undefined.

3. An optional decimal digit string that specifies the minimum field width. If the converted value has fewer characters than the field width, the field is padded on the left to the length specified by the field width. If the left-adjustment option is specified, the field is padded on the right. If the result of a conversion is wider than the field width, the field is expanded to contain the converted result. No truncation occurs. However, a small precision may cause truncation on the right.
4. An optional precision. The precision is a . (dot) followed by a decimal digit string. If no precision is given, it is treated as 0 (zero). The precision specifies:
   - The minimum number of digits to be displayed for the **d**, **o**, **i**, **u**, **x**, or **X** conversions.
   - The number of digits to be displayed after the radix character for the **e** and **f** conversions.
   - The maximum number of significant digits for the **g** conversion.
   - The maximum number of bytes to be printed from a string in the **s** conversion.
5. A character that indicates the type of conversion to be applied, such as:

% Performs no conversion. Prints a % (percent sign).

d, **i** Accepts an integer value and converts it to signed decimal notation. The precision specifies the minimum number of digits to be displayed. If the value being converted can be represented in fewer digits, it is expanded with leading zeros. The default precision is 1. The result of converting a zero value with a precision of zero is a null string. Specifying a field width with a zero as a leading character causes the field width value to be padded with leading zeros.

o Accepts an integer value and converts it to signed octal notation. The precision specifies the minimum number of digits to be displayed. If the value being converted can be represented in fewer digits, it is expanded with leading zeros. The default precision is 1. The result of converting a zero value with a precision of zero is a null string. Specifying a field width with a zero as a leading character causes the field width value to be padded with leading zeros. An octal value for field width is not implied.

u Accepts an integer value and converts it to unsigned decimal notation. The precision specifies the minimum number of digits to be displayed. If the value being converted can be represented in fewer digits, it is expanded with leading zeros. The default precision is 1. The result of converting a zero value with a precision of zero is a null string. Specifying a field width with a zero as a leading character causes the field width value to be padded with leading zeros.

x, **X** Accepts an integer value and converts it to hexadecimal notation. The letters abcdef are used for the **x** conversion and the letters ABCDEF are used for the **X** conversion. The precision specifies the minimum number of digits to be displayed. If the value being converted can be represented in fewer digits, it is expanded with leading zeros. The default precision is 1. The result of converting a zero value with a precision of zero is a null string. Specifying a field width with a zero as a leading character causes the field width value to be padded with leading zeros.

f Accepts a float or double value and converts it to decimal notation in the format [**-**] *ddd***.***ddd*. The number of digits after the radix character (shown here as the decimal point) is equal to the precision specification. The **LC_NUMERIC** locale category determines the radix character to use tin this format. If no precision is specified, then six digits are output. If the precision is `0` (zero), then no radix character will be displayed.

e, **E** Accepts a float or double value and converts it to the exponential form [**-**] *d***.***dd***e**{**+**|**-**}*dd*. There is one digit before the radix character (shown here as the decimal point) and the number of digits after the radix character is equal to the precision specification. The **LC_NUMERIC** locale category determines the radix character to use tin this format. If no precision is specified, then six digits are output. If the precision is 0 (zero), then no radix character will be displayed. The **E** conversion character produces a number with E instead of e before the exponent. The exponent always contains at least two digits. However, if the value to be printed requires an exponent greater than two digits, additional exponent digits are printed as necessary.

g, **G** Accepts a float or double value and converts it in the style of the **f** or **e** conversion characters (or **E** in the case of the **G** conversion), with the precision specifying the number of significant digits. Trailing zeros are removed from the result. A radix character is displayed only if it is followed by a digit. The style used depends on the value converted. Style **g** results only if the exponent resulting from the conversion is less than -4, or if it is greater than or equal to the precision.

c Accepts a value as a string and prints the first character in the string.

s Accepts a value as a string and prints characters from the string until the end of the string is encountered or the number of characters indicated by the precision is reached. If no precision is specified, all characters up to the first null character are printed.

b Accepts a value as a string, that may contain backslash-escape sequences. Bytes from the converted string are printed until the end of the string or number of bytes indicated by the precision specification is reached. If the precision is omitted, all bytes until the first null character are printed.

The following backslash-escape sequences are supported:

- The escape sequences previously listed above under the description of the *Format* parameter. These are converted to the individual characters they represented.
- The \c (backslash c) sequence, which is not displayed and causes the **printf** command to ignore any remaining characters in the string parameter containing it, any remaining string parameters, and any additional characters in the *Format* parameter.

Exit Status

This command returns the following exit values:

0 Successful completion.

>0 An error occurred.

Examples

1. Enter the following command:

   ```
   printf "%5d%4d\n" 1 21 321 4321 54321
   ```

   This produces the following output:

   ```
      1  21
    3214321
   54321   0
   ```

   The *Format* parameter is used three times to print all of the given strings. The 0 (zero) is supplied by the **printf** command to satisfy the last `%4d` conversion specification.

2. Enter the following command:

   ```
   printf "%c %c" 78 79
   ```

   This produces the following output:

   ```
   7 7
   ```

Files

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

Related Information

The **/usr/bin/echo** command.

The **printf** subroutine in *AIX 5L Version 5.2 Technical Reference: Base Operating System and Extensions*.

The Input and Output Handling Programmer's Overview in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs* describes the files, commands, and subroutines used for low-level, stream, terminal, and asynchronous I/O interfaces.

The National Language Support Overview in *AIX 5L Version 5.2 National Language Support Guide and Reference*.

---

proccred Command

Purpose

Prints the credentials (effective, real, saved user IDs and group IDs) of processes.

Syntax

proccred [*ProcessID*] ...

Description

The **/proc** filesystem provides a mechanism to control processes. It also gives access to information about the current state of processes and threads, but in binary form. The proctools commands provide ascii reports based on some of the available information.

Most of the commands take a list of process IDs or **/proc/***ProcessID* strings as input. The shell expansion **/proc/*** can therefore be used to specify all processes in the system.

Each of the proctools commands gathers information from **/proc** for the specified processes and displays it to the user. The proctools commands like procrun and procstop start and stop a process using the **/proc** interface.

The information gathered by the commands from **/proc** is a snapshot of the current state of processes, and therefore can vary at any instant except for stopped processes.

The **proccred** command prints the credentials (effective, real, saved user IDs and group IDs) of processes.

Flags

ProcessID	Specifies the process id.

Examples

1. To display the credentials of process 5046, enter:

```
proccred  5046
```

Files

/proc	Contains the **/proc** filesystem.

Related Information

The **procfiles** command, **procflags** command, **procldd** command, **procmap** command, **procrun** command, **procsig** command, **procstack** command, **procstop** command, **proctree** command, **procwait** command, **procwdx** command.

---

processPtf Command

Purpose

Generates the *option***.prereq** file and **lpp.doc** file for each PTF.

Syntax

processPtf [**-v** *Version* **-r** *Release* **-m** *Mod* **-f** *Fix*]

Description

The **processPtf** command also generates several data files in the PTF directory. The **processPtf** command is invoked by the **ptfpkg** command to generate PTF specific files. The **lpp.doc** file is archived into **liblpp.a** for the update. The *option***.prereq** file is included in the **lpp_name** file for the update. The **processPtf** command also generates the **filenamesList**, **ifreqsList**, **coreqsList**, **aparsList**, **prereqsList**, and **supersedesList** files in the PTF directory. The list files are data files and are not archived into **liblpp.a**.

The list files are generated by combining the information from the most recent fileset PTF with the information for the current PTF. The **$TOP/UPDATE/$OPTION/ptfsList** file identifies all PTFs since the last cum for the fileset. The last PTF in this file is assumed to be the most recent fileset PTF. The **processPtf** command generates the *option***.prereq file** from the new **prereqsList**, **coreqsList**, and **ifreqsList** information. If the most recent cum PTF listed in the **$TOP/UPDATE/$OPTION/cumsList** file has a non-zero fix level, then **processPtf** puts an explicit prereq in the *option***.prereq** file for the most recent cum PTF.

The **processPtf** command searches the **$TOP/HISTORY/memo_info** file for the APAR text for each APAR in the **aparsList** file and generates the **lpp.doc** file.

Flags

Note: The **-v**, **-r**, **-m**, and **-f** flags are only used to override the default version, release, mod, and fix level values. These flags are usually only used if the VRMF has been overridden in the **ptf_pkg.$BLDCYCLE** file which is used as input to the **ptfpkg** command.

-f *Fix*	Specifies the fix level.
-m *Mod*	Specifies the modification level.
-r *Release*	Specifies the release number.
-v *Version*	Specifies the version number.

Environmental Variables

BLDCYCLE	Specifies the current build cycle. The build cycle identifier identifies the build in which the PTF was created.
BUILD_TYPE	Specifies the type of environment this build is occurring in. Valid values are production, area, and sandbox, where sandbox builds are development level builds. The **ptfpkg** command requires valid APAR text in the **$TOP/HISTORY/abstracts** file for each APAR if the build type is production.
TOP	Specifies that the **$TOP** environment variable is set to the base of the update tree. The **PTF**, **HISTORY**, and **UPDATE** directories exist relative to **$TOP**.

Examples

1. To invoke the **processPtf** command to generate the **lpp.doc**, *option***.prereq**, and list files for the current PTF using the default version, release, mod, and fix levels, type:

   ```
   processPtf
   ```

2. To invoke the **processPtf** command to generate PTF information for a cum PTF with a non-zero fix level value, type:

   ```
   processPtf -v 4 -r 2 -m1 -f3
   ```

Input Files

cumsList

The **$TOP/UPDATE/$OPTION/cumsList** file is created and maintained by the **ptfpkg** command and is not provided by the user. The **cumsList** file contains the list of successfully built cumulative PTFs since the base level release. The format of this file is one PTF ID per line.

memo_info

The **$TOP/HISTORY/memo_info** file is optional. It contains information by APAR number that is included in the **lpp.doc** file and displayed to the user when the PTF is installed. Its format is:

```
APARID
TEXT
@
Example:
IX35837
```

After applying this PTF, the system must be rebooted. For diskless clients, the boot image of the diskless client has to be rebuilt and the diskless client rebooted for this fix to take effect. @ IX35995 Should @ IX35995 be here?

> **Note:** Please read the file **/usr/lpp/bos/README.M-ACPA** for details and restrictions of the Multimedia Audio Capture and Playback Adapter (M-ACPA) for playing two monaural tracks simultaneously. @ Should @ be here?

ptfsList	The **$TOP/UPDATE/$OPTION/ptfsList** file is created and maintained by the **ptfpkg** command and is not provided by the user. The **ptfsList** file contains the list of fileset PTFs successfully built since the most recent cum PTF. The format of this file is one PTF ID per line.
wk_ptf_pkg	The **$TOP/UPDATE/$OPTION/$PTF/wk_ptf_pkg** file is created by the **ptfpkg** command from the **$TOP/UPDATE/$LPP/ptf_pkg.$BLDCYCLE** file. It contains the information for only the PTF currently being processed.

Output Files

aparsList/filenamesList/ifreqsList/coreqsList/ prereqsList/supersedesList	The list files are generated in the **$TOP/UPDATE/$OPTION/$PTF** directory. The list files from the most recent fileset PTF are combined with the new information from the **wk_ptf_pkg** file to generate the list files for the current PTF.
lpp.doc	The **lpp.doc** file is generated if the **$TOP/HISTORY/memo_info** file exists and contains APAR information for one or more APARs in the **aparsList** file. This file is archived into **liblpp.a**.
option.**prereq**	The *option*.**prereq** file contains the ifreq, coreq, and prereq data for the current PTF. This file is copied into the **lpp_name** file for the update.

Related Information

The **ptfpkg** command.

---

procfiles Command

Purpose

Reports information about all file descriptors opened by processes.

Syntax

procfiles [**-F**] [**-n**][*ProcessID*] ...

Description

The **/proc** filesystem provides a mechanism to control processes. It also gives access to information about the current state of processes and threads, but in binary form. The proctools commands provide ascii reports based on some of the available information.

Most of the commands take a list of process IDs or **/proc/***ProcessID* strings as input. The shell expansion **/proc/*** can therefore be used to specify all processes in the system.

Each of the proctools commands gathers information from **/proc** for the specified processes and displays it to the user. The proctools commands like procrun and procstop start and stop a process using the **/proc** interface.

The information gathered by the commands from **/proc** is a snapshot of the current state of processes, and therefore can vary at any instant except for stopped processes.

The **procfiles** command reports information on all file descriptors opened by processes. With the **-n** option it also displays the names of the corresponding files.

Flags

-F	Forces procfiles to take control of the target process even if another process has control.
-n	Prints the names of the files referred to by file descriptors.
ProcessID	Specifies the process id.

Examples

1. To display status and control information on the file descriptors opened by process 11928, enter:

   ```
   procfiles 11928
   ```

   The output of this command might look like this:

   ```
   11928 : -sh
     Current rlimit: 2000 file descriptors
      0: S_IFCHR mode:0622 dev:10,4  ino:2584 uid:100 gid:100 rdev:28,1
         O_RDONLY
      1: S_IFCHR mode:0622 dev:10,4  ino:2584 uid:100 gid:100 rdev:28,1
         O_RDONLY
      2: S_IFCHR mode:0622 dev:10,4  ino:2584 uid:100 gid:100 rdev:28,1
         O_RDONLY
     63: S_IFREG mode:0600 dev:10,8  ino:311 uid:100 gid:100 rdev:40960,10317
         O_RDONLY size:2574
   ```

2. To display name, status and control information on the file descriptors opened by process 15502, enter:

   ```
   procfiles -n 15502
   ```

   The output of this command might look like this:

   ```
   15502 : /home/guest/test
     Current rlimit: 2000 file descriptors
      0: S_IFCHR mode:0622 dev:10,4  ino:2584 uid:100 gid:100 rdev:28,1
         O_RDONLY
      1: S_IFCHR mode:0622 dev:10,4  ino:2584 uid:100 gid:100 rdev:28,1
         O_RDONLY
      2: S_IFCHR mode:0622 dev:10,4  ino:2584 uid:100 gid:100 rdev:28,1
         O_RDONLY
      3: S_IFREG mode:0644 dev:10,7  ino:26 uid:100 gid:100 rdev:0,0
         O_RDONLY size:0  name:/tmp/foo
   ```

Files

/proc	Contains the **/proc** filesystem.

Related Information

The **proccred** command, **procflags** command, **procldd** command, **procmap** command, **procrun** command, **procsig** command, **procstack** command, **procstop** command, **proctree** command, **procwait** command, **procwdx** command.

---

procflags Command

Purpose

Prints the **/proc** tracing flags, the **pending** and **held** signals, and other **/proc** status information for each thread in the specified processes.

Syntax

procflags [**-r**] [*ProcessID*] ...

Description

The **/proc** filesystem provides a mechanism to control processes. It also gives access to information about the current state of processes and threads, but in binary form. The proctools commands provide ascii reports based on some of the available information.

Most of the commands take a list of process IDs or **/proc/***ProcessID* strings as input. The shell expansion **/proc/*** can therefore be used to specify all processes in the system.

Each of the proctools commands gathers information from **/proc** for the specified processes and displays it to the user. The proctools commands like procrun and procstop start and stop a process using the **/proc** interface.

The information gathered by the commands from **/proc** is a snapshot of the current state of processes, and therefore can vary at any instant except for stopped processes.

The **procflags** command prints the **/proc** tracing flags, the pending and held signals, and other **/proc** status information for each thread in the specified processes. The machine register contents are printed when option **-r** is used and the process is stopped on an event of interest. The events of interest are **PR_REQUESTED**, **PR_FAULTED**, **PR_SYSENTRY**, and **PR_SYSEXIT** as defined in **<sys/procfs.h>**.

Flags

-r	Displays the current machine registers state if a process is stopped in an event of interest.
ProcessID	Specifies the process id.

Examples

1. To display the tracing flags of process 5046, enter:

   ```
   procflags  5046
   ```

   The output of this command might look like this:

   ```
   5046 : -sh
   data model = _ILP32 flags = PR_FORK
   /4289: flags = PR_ASLEEP | PR_NOREGS
   ```

2. To display the tracing flags and registers values of process 5040 which was stopped on an event of interest, enter:

   ```
   procflags -r 5040
   ```

   The output of this command might look like this:

   ```
   5040 : ls
   data model = _ILP32 flags = PR_FORK
   /6999: flags = PR_STOPPED | PR_ISTOP
   why = PR_FAULTED  what = FLTBPT what = kfork
   gpr0  = 0x0               gpr1  = 0x2ff227b0        gpr2  = 0xf0083bec
   gpr3  = 0x2ff22cb3        gpr4  = 0x11              gpr5  = 0x65
   gpr6  = 0x50              gpr7  = 0x0               gpr8  = 0x41707a7c
   gpr9  = 0x4c4f47          gpr10 = 0x80000000        gpr11 = 0x34e0
   gpr12 = 0x0               gpr13 = 0xdeadbeef        gpr14 = 0x1
   gpr15 = 0x2ff22c0c        gpr16 = 0x2ff22c14        gpr17 = 0x0
   gpr18 = 0xdeadbeef        gpr19 = 0xdeadbeef        gpr20 = 0xdeadbeef
   gpr21 = 0xdeadbeef        gpr22 = 0x10              gpr23 = 0xfd
   gpr24 = 0x2f              gpr25 = 0x2ff227f0        gpr26 = 0x0
   ```

```
gpr27 = 0x2ff22d87      gpr28 = 0x2ff22cb3      gpr29 = 0x0
gpr30 = 0x0             gpr31 = 0xf0048260      iar = 0xd01be900
msr = 0x2d032           cr = 0x28222442         lr = 0xd01d9de0
ctr = 0xec              xer = 0x0               fpscr = 0x0
fpscrx = 0x0
```

Files

/proc	Contains the **/proc** filesystem.

Related Information

The **proccred** command, **procfiles** command, **procldd** command, **procmap** command, **procrun** command, **procsig** command, **procstack** command, **procstop** command, **proctree** command, **procwait** command, **procwdx** command.

---

procldd Command

Purpose

Lists the dynamic libraries loaded by processes, including shared objects explicitly attached using **dlopen()**.

Syntax

procldd [**-F**] [*ProcessID*] ...

Description

The **/proc** filesystem provides a mechanism to control processes. It also gives access to information about the current state of processes and threads, but in binary form. The proctools commands provide ascii reports based on some of the available information.

Most of the commands take a list of process IDs or **/proc/***ProcessID* strings as input. The shell expansion **/proc/*** can therefore be used to specify all processes in the system.

Each of the proctools commands gathers information from **/proc** for the specified processes and displays it to the user. The proctools commands like procrun and procstop start and stop a process using the **/proc** interface.

The information gathered by the commands from **/proc** is a snapshot of the current state of processes, and therefore can vary at any instant except for stopped processes.

The **procldd** command lists the dynamic libraries loaded by processes, including shared objects explicitly attached using **dlopen()**. All the information needed is gathered from the **/proc/***ProcessID***/map** files.

Flags

-F	Forces procldd to take control of the target process even if another process has control.
ProcessID	Specifies the process id.

Examples

1. To display the list of dynamic libraries loaded by process 11928, enter:

   ```
   procldd 11928
   ```

The output of this command might look like this:

```
11928 : -sh
/usr/lib/nls/loc/en_US
/usr/lib/libcrypt.a
/usr/lib/libc.a
```

Files

/proc	Contains the **/proc** filesystem.

Related Information

The **proccred** command, **procfiles** command, **procflags** command, **procmap** command, **procrun** command, **procsig** command, **procstack** command, **procstop** command, **proctree** command, **procwait** command, **procwdx** command.

---

procmap Command

Purpose

Prints the address space map of processes.

Syntax

procmap [**-F**] [*ProcessID*] ...

Description

The **/proc** filesystem provides a mechanism to control processes. It also gives access to information about the current state of processes and threads, but in binary form. The proctools commands provide ascii reports based on some of the available information.

Most of the commands take a list of process IDs or **/proc/***ProcessID* strings as input. The shell expansion **/proc/*** can therefore be used to specify all processes in the system.

Each of the proctools commands gathers information from **/proc** for the specified processes and displays it to the user. The proctools commands like procrun and procstop start and stop a process using the **/proc** interface.

The information gathered by the commands from **/proc** is a snapshot of the current state of processes, and therefore can vary at any instant except for stopped processes.

The **procmap** command prints the address space map of processes. It displays the starting address and size of each of the mapped segments in the process. It gets all the information necessary from the **/proc/***ProcessID***/map** files.

Flags

-F	Forces procmap to take control of the target process even if another process has control.
ProcessID	Specifies the process id.

Examples

1. To display the address space of process 11928, enter:

   ```
   procmap 11928
   ```

The output of this command might look like this:

```
11928 : -sh
10000000         235K  read/exec         sh
20000ca0          31K  read/write        sh
d0072000          11K  read/exec         /usr/lib/nls/loc/en_US
d0075130           8K  read/write        /usr/lib/nls/loc/en_US
d00710f8           2K  read/exec         /usr/lib/libcrypt.a
f00c5530           0K  read/write        /usr/lib/libcrypt.a
d01ba240        1895K  read/exec         /usr/lib/libc.a
f0042220         519K  read/write        /usr/lib/libc.a
   Total        2705K
```

Files

/proc	Contains the **/proc** filesystem.

Related Information

The **proccred** command, **procfiles** command, **procflags** command, **procldd** command, **procrun** command, **procsig** command, **procstack** command, **procstop** command, **proctree** command, **procwait** command, **procwdx** command.

---

procrun Command

Purpose

Starts a process that has stopped on the **PR_REQUESTED** event.

Syntax

procrun [*ProcessID*] ...

Description

The **/proc** filesystem provides a mechanism to control processes. It also gives access to information about the current state of processes and threads, but in binary form. The proctools commands provide ascii reports based on some of the available information.

Most of the commands take a list of process IDs or **/proc/***ProcessID* strings as input. The shell expansion **/proc/*** can therefore be used to specify all processes in the system.

Each of the proctools commands gathers information from **/proc** for the specified processes and displays it to the user. The proctools commands like procrun and procstop start and stop a process using the **/proc** interface.

The information gathered by the commands from **/proc** is a snapshot of the current state of processes, and therefore can vary at any instant except for stopped processes.

The **procrun** command starts the process that has stopped on the **PR_REQUESTED** event.

Flags

ProcessID	Specifies the process id.

Examples

1. To restart process 30192 which was stopped on the **PR_REQUESTED** event, enter:

   ```
   procrun 30192
   ```

Files

/proc	Contains the **/proc** filesystem.

Related Information

The **proccred** command, **procfiles** command, **procflags** command, **procldd** command, **procmap** command, **procsig** command, **procstack** command, **procstop** command, **proctree** command, **procwait** command, **procwdx** command.

---

procsig Command

Purpose

Lists the signal actions defined by processes.

Syntax

procsig [*ProcessID*] ...

Description

The **/proc** filesystem provides a mechanism to control processes. It also gives access to information about the current state of processes and threads, but in binary form. The proctools commands provide ascii reports based on some of the available information.

Most of the commands take a list of process IDs or **/proc/***ProcessID* strings as input. The shell expansion **/proc/*** can therefore be used to specify all processes in the system.

Each of the proctools commands gathers information from **/proc** for the specified processes and displays it to the user. The proctools commands like procrun and procstop start and stop a process using the **/proc** interface.

The information gathered by the commands from **/proc** is a snapshot of the current state of processes, and therefore can vary at any instant except for stopped processes.

The **procsig** command lists the signal actions defined by processes.

Flags

ProcessID	Specifies the process id.

Examples

1. To list all the signal actions defined for process 11928, enter:

   ```
   procsig 11928
   ```

   The output of this command might look like this:

   ```
   HUP         caught
   INT         caught
   QUIT        caught
   ```

```
ILL         caught
TRAP        caught
ABRT        caught
EMT         caught
FPE         caught
KILL        default  RESTART
BUS         caught
SEGV        default
SYS         caught
PIPE        caught
ALRM        caught
TERM        ignored
URG         default
STOP        default
TSTP        ignored
CONT        default
CHLD        default
TTIN        ignored
TTOU        ignored
IO          default
XCPU        default
XFSZ        ignored
MSG         default
WINCH       default
PWR         default
USR1        caught
USR2        caught
PROF        default
DANGER      default
VTALRM      default
MIGRATE     default
PRE         default
VIRT        default
ALRM1       default
WAITING     default
CPUFAIL     default
KAP         default
RETRACT     default
SOUND       default
SAK         default
```

Files

/proc	Contains the **/proc** filesystem.

Related Information

The **proccred** command, **procfiles** command, **procflags** command, **procldd** command, **procmap** command, **procrun** command, **procstack** command, **procstop** command, **proctree** command, **procwait** command, **procwdx** command.

---

procstack Command

Purpose

Prints the hexadecimal addresses and symbolic names for each of the stack frames of the current thread in processes.

Syntax

procstack [**-F**] [*ProcessID*] ...

Description

The **/proc** filesystem provides a mechanism to control processes. It also gives access to information about the current state of processes and threads, but in binary form. The proctools commands provide ascii reports based on some of the available information.

Most of the commands take a list of process IDs or **/proc/***ProcessID* strings as input. The shell expansion **/proc/*** can therefore be used to specify all processes in the system.

Each of the proctools commands gathers information from **/proc** for the specified processes and displays it to the user. The proctools commands like procrun and procstop start and stop a process using the **/proc** interface.

The information gathered by the commands from **/proc** is a snapshot of the current state of processes, and therefore can vary at any instant except for stopped processes.

The **procstack** command prints the hexadecimal addresses and symbolic names for each of the stack frames of the current thread in processes.

Flags

-F	Forces procstack to take control of the target process even if another process has control.
ProcessID	Specifies the process id.

Examples

1. To display the current stack of process 11928, enter:

   ```
   procstack 11928
   ```

   The output of this command might look like this:

   ```
   11928 : -sh
   d01d15c4  waitpid   (?, ?, ?) + e0
   10007a1c  job_wait   (?) + 144
   10020298  xec_switch   (?, ?, ?, ?, ?) + 9c0
   10021db4  sh_exec   (?, ?, ?) + 304
   10001370  exfile   () + 628
   10000300  main   (?, ?) + a1c
   10000100  __start   () + 8c
   ```

Files

/proc	Contains the **/proc** filesystem.

Related Information

The **proccred** command, **procfiles** command, **procflags** command, **procldd** command, **procmap** command, **procrun** command, **procsig** command, **procstop** command, **proctree** command, **procwait** command, **procwdx** command.

---

procstop Command

Purpose

Stops processes on the **PR_REQUESTED** event.

Syntax

procstop [*ProcessID*] ...

Description

The **/proc** filesystem provides a mechanism to control processes. It also gives access to information about the current state of processes and threads, but in binary form. The proctools commands provide ascii reports based on some of the available information.

Most of the commands take a list of process IDs or **/proc/***ProcessID* strings as input. The shell expansion **/proc/*** can therefore be used to specify all processes in the system.

Each of the proctools commands gathers information from **/proc** for the specified processes and displays it to the user. The proctools commands like procrun and procstop start and stop a process using the **/proc** interface.

The information gathered by the commands from **/proc** is a snapshot of the current state of processes, and therefore can vary at any instant except for stopped processes.

The **procstop** command stops processes on the **PR_REQUESTED** event.

Flags

ProcessID	Specifies the process id.

Examples

1. To stop process 7500 on the **PR_REQUESTED** event, enter:

   ```
   procstop 7500
   ```

Files

/proc	Contains the **/proc** filesystem.

Related Information

The **proccred** command, **procfiles** command, **procflags** command, **procldd** command, **procmap** command, **procrun** command, **procsig** command, **procstack** command, **proctree** command, **procwait** command, **procwdx** command.

---

proctree Command

Purpose

Prints the process tree containing the specified process IDs or users.

Syntax

proctree [**-a**] [{ *ProcessID* | *User* }]

Description

The **/proc** filesystem provides a mechanism to control processes. It also gives access to information about the current state of processes and threads, but in binary form. The proctools commands provide ascii reports based on some of the available information.

Most of the commands take a list of process IDs or **/proc/***ProcessID* strings as input. The shell expansion **/proc/*** can therefore be used to specify all processes in the system.

Each of the proctools commands gathers information from **/proc** for the specified processes and displays it to the user. The proctools commands like procrun and procstop start and stop a process using the **/proc** interface.

The information gathered by the commands from **/proc** is a snapshot of the current state of processes, and therefore can vary at any instant except for stopped processes.

The **proctree** command prints the process tree containing the specified process IDs or users. The child processes are indented from their respective parent processes. An argument of all digits is taken to be a process ID, otherwise it is assumed to be a user login name. The default action is to report on all processes, except children of process 0.

Flags

-a	Include children of process 0 in the display. The default is to exclude them.
ProcessID	Specifies the process id.
User	Specifies the User.

Examples

1. To display the ancestors and all the children of process 12312, enter:

   ```
   proctree 12312
   ```

   The output of this command might look like this:

   ```
   4954    /usr/sbin/srcmstr
      7224    /usr/sbin/inetd
         5958    telnetd -a
            13212    -sh
               14718    ./proctree 13212
   ```

2. To display the ancestors and children of process 12312, including children of process 0, enter:

   ```
   proctree -a 12312
   ```

   The output of this command might look like this:

   ```
   1    /etc/init
      4954    /usr/sbin/srcmstr
         7224    /usr/sbin/inetd
            5958    telnetd -a
               13212    -sh
                  14724    ./proctree -a 13212
   ```

Files

/proc	Contains the **/proc** filesystem.

Related Information

The **proccred** command, **procfiles** command, **procflags** command, **procldd** command, **procmap** command, **procrun** command, **procsig** command, **procstack** command, **procstop** command, **procwait** command, **procwdx** command.

procwait Command

Purpose

Waits for all of the specified processes to terminate.

Syntax

procwait [**-v**] [*ProcessID*] ...

Description

The **/proc** filesystem provides a mechanism to control processes. It also gives access to information about the current state of processes and threads, but in binary form. The proctools commands provide ascii reports based on some of the available information.

Most of the commands take a list of process IDs or **/proc/***ProcessID* strings as input. The shell expansion **/proc/*** can therefore be used to specify all processes in the system.

Each of the proctools commands gathers information from **/proc** for the specified processes and displays it to the user. The proctools commands like procrun and procstop start and stop a process using the **/proc** interface.

The information gathered by the commands from **/proc** is a snapshot of the current state of processes, and therefore can vary at any instant except for stopped processes.

The **procwait** command waits for all of the specified processes to terminate.

Flags

-v	Specifies verbose output. Reports terminations to standard output.
ProcessID	Specifies the process id.

Examples

1. To wait for process 12942 to exit and display the status, enter:

   ```
   procwait -v 12942
   ```

   The output of this command might look like this:

   ```
   12942 : terminated, exit status 0
   ```

Files

/proc	Contains the **/proc** filesystem.

Related Information

The **proccred** command, **procfiles** command, **procflags** command, **procldd** command, **procmap** command, **procrun** command, **procsig** command, **procstack** command, **procstop** command, **proctree** command, **procwdx** command.

procwdx Command

Purpose

Prints the current working directory of processes.

Syntax

procwdx [**-F**] [*ProcessID*] ...

Description

The **/proc** filesystem provides a mechanism to control processes. It also gives access to information about the current state of processes and threads, but in binary form. The proctools commands provide ascii reports based on some of the available information.

Most of the commands take a list of process IDs or **/proc/***ProcessID* strings as input. The shell expansion **/proc/*** can therefore be used to specify all processes in the system.

Each of the proctools commands gathers information from **/proc** for the specified processes and displays it to the user. The proctools commands like procrun and procstop start and stop a process using the **/proc** interface.

The information gathered by the commands from **/proc** is a snapshot of the current state of processes, and therefore can vary at any instant except for stopped processes.

The **procwdx** command prints the current working directory of processes.

Flags

-F	Forces procfiles to take control of the target process even if another process has control.
ProcessID	Specifies the process id.

Examples

1. To display the current working directory of process 11928, enter:

   ```
   procwdx 11928
   ```

   The output of this command might look like this:

   ```
   11928 :  /home/guest
   ```

Files

/proc	Contains the **/proc** filesystem.

Related Information

The **proccred** command, **procfiles** command, **procflags** command, **procldd** command, **procmap** command, **procrun** command, **procsig** command, **procstack** command, **procstop** command, **proctree** command, **procwait** command.

prof Command

Purpose

Displays object file profile data.

Syntax

prof [**-t** | **-c** | **-a** | **-n**] [**-o** | **-x**] [**-g**] [**-z**] [**-h**] [**-s**] [**-S**] [**-v**] [**-L** *PathName*] [*Program*] [**-m** *MonitorData* ...]

Description

The **prof** command interprets profile data collected by the **monitor** subroutine for the object file *Program* (**a.out** by default). It reads the symbol table in the object file *Program* and correlates it with the profile file (**mon.out** by default). The **prof** command displays, for each external text symbol, the percentage of execution time spent between the address of that symbol and the address of the next, the number of times that function was called, and the average number of milliseconds per call.

> **Note:** Symbols from C++ object files have their names demangled before they are used.

To tally the number of calls to a function, you must have compiled the file using the **cc** command with the **-p** flag. The **-p** flag causes the compiler to insert a call to the **mcount** subroutine into the object code generated for each recompiled function of your program. While the program runs, each time a parent calls a child function the child calls the **mcount** subroutine to increment a distinct counter for that parent-child pair. Programs not recompiled with the **-p** flag do not have the **mcount** subroutine inserted and therefore keep no record of which function called them.

The **-p** flag also arranges for the object file to include a special profiling startup function that calls the **monitor** subroutine when the program begins and ends. The call to the **monitor** subroutine when the program ends actually writes the **mon.out** file. Therefore, only programs that explicitly exit or return from the main program cause the **mon.out** file to be produced.

The location and names of the objects loaded are stored in the **mon.out** file. If you do not select any flags, **prof** will use these names. You must specify a program or use the **-L** option to access other objects.

> **Note:** Imported external routine calls, such as a call to a shared library routine, have an intermediate call to local **glink** code that sets up the call to the actual routine. If the timer clock goes off while running this code, time is charged to a routine called *routine***.gl**, where *routine* is the routine being called. For example, if the timer goes off while in the **glink** code to call the **printf** subroutine, time is charged to the **printf.gl** routine.

Flags

The mutually exclusive flags **a**, **c**, **n**, and **t** determine how the **prof** command sorts the output lines:

-a Sorts by increasing symbol address.
-c Sorts by decreasing number of calls.
-n Sorts lexically by symbol name.
-t Sorts by decreasing percentage of total time (default).

> **Note:** The **prof** command can still run successfully if you use more than one of flags **a**, **c**, **n**, and **t** in the same command. The **prof** command accepts the first of these flags it encounters on the command line and ignores the others.

The mutually exclusive flags **o** and **x** specify how to display the address of each symbol monitored.

-o Displays each address in octal, along with the symbol name.
-x Displays each address in hexadecimal, along with the symbol name.

> **Note:** The **prof** command can still run successfully if you use both the **-o** and **-x** flags in the same command. The **prof** command accepts the first of these two flags it encounters on the command line and ignores the other flag.

Use the following flags in any combination:

-g	Includes non-global symbols (static functions).
-h	Suppresses the heading normally displayed on the report. This is useful if the report is to be processed further.
-L *PathName*	Uses alternate path name for locating shared objects.
-m *MonitorData*	Takes profiling data from *MonitorData* instead of **mon.out**.
-s	Produces a summary file in **mon.sum**. This is useful when more than one profile file is specified.
-S	Displays a summary of monitoring parameters and statistics on standard error.
-v	Suppresses all printing and sends a graphic version of the profile to standard output for display by the plot filters. When plotting, low and high numbers, by default 0 and 100, can be given to cause a selected percentage of the profile to be plotted with accordingly higher resolution.
-z	Includes all symbols in the profile range, even if associated with 0 (zero) calls and 0 (zero) time.

Examples

1. To display, without a header, the amount of time spent at each symbol address, sorted by time, enter:
   ```
   prof -t -h
   ```
2. The following example obtains a local version of any shared libraries used to create the **runfile** file in the **/home/score/lib** directory. The data file used will be **runfile.mon** rather than **mon.out**.
   ```
   prof -x -L/home/score/lib runfile -m runfile.mon
   ```

Files

mon.out	Default profile.
a.out	Default object file.
mon.sum	Summary profile.

Related Information

The **cc** command, **nm** command, **gprof** command.

The **exit** subroutine, **profil** subroutine, **monitor** subroutine.

The Commands Overview in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices*.

The Subroutines Overview in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

proff Command

Purpose

Formats text for printers with personal printer data streams.

Syntax

proff [**-L***List*] [**-P***Printer*] [**-t**] [*nroffFlags*] [*File* ...]

Description

The **proff** command formats text by using the **nroff** command on the specified files for printers that support ppds (personal printer data streams), such as the Quietwriter III printer, the Quickwriter printer, and the Proprinter printer.

If no file is specified, standard input is read. A parameter value of **-** (minus) specifies standard input.

Parameters

nroffFlags	Specifies the **nroff** command flags used by the **proff** command to format the text file for a ppds-supported printer output.
File	Specifies the text file that the **proff** command formats for printers that support ppds.

Flags

-L*List* Passes the specified list as flags for the **qprt** command.

To pass a single flag to the **qprt** command, use the **-L** flag followed immediately by the **nroff** command flag being passed. For example:

```
-L-h.
```

To pass multiple flags or a string to the **lpr** command, use the **-L** flag followed immediately by the flags or string enclosed by ″ ″ (double quotes):

```
-L"-h -r -m".
```

-P*Printer* Sends output to a specified printer corresponding to an entry in the **/etc/qconfig** file. The default is taken from the **PRINTER** environment variable, if it exists; otherwise the system default queue name is used.

-t Sends output to standard output.

- Specifies that standard input is used as the source for the formatting process.

All other flags are passed to the **nroff** command.

Example

The following is a typical command sequence to process output for the IBM Proprinter printer:

```
proff -t testfile
```

Environment Variable

PRINTER Specifies the desired printer queue.

Files

/usr/share/lib/nterm/tab.ppds	Contains driving tables for printers with personal printer data streams.
/etc/qconfig	Describes the queues and devices.

Related Information

The **col** command, **eqn** command, **lpr** command, **nroff** command, **qprt** command, **tbl** command.

---

prompter Command

Purpose

Starts a prompting editor.

Syntax

prompter [**-erase** *Character*] [**-kill** *Character*] [**-prepend** | **-noprepend**] [**-rapid** | **-norapid**] *File*

Description

Part of the Message Handler (MH) package, the **prompter** command starts the prompting editor for message entry. The **prompter** command is not started by the user. The **prompter** command is called by other programs only.

The **prompter** command opens the file specified by the *File* parameter, scans it for empty components such as the `To:` component, and prompts you to fill in the blank fields. If you press the Enter key without filling in a required field, the **prompter** command deletes the component.

The **prompter** command accepts text for the body of the message after the first blank line or line of dashes in the file. If the body already contains text and the **-noprepend** flag is specified, the **prompter** command displays the text followed by the message:

```
--------Enter additional text
```

The **prompter** command appends any new text entered after the existing message. If you specify the **-prepend** flag, the **prompter** command displays the following message:

```
--------Enter initial text
```

Any new text precedes the body of the original message. When you press the Ctrl-D key sequence for End of File, the **prompter** command ends text entry and returns control to the calling program.

Flags

-erase *Character*	Sets the character to be used as the erase character. The value of the *Character* variable can be the octal representation of the character in the form `\NNN` where `\NNN` is a number or the character itself. For example, the character `\e` is `\145` in octal representation.
-help	Lists the command syntax, available switches (toggles), and version information. **Note:** For MH, the name of this flag must be fully spelled out.
-kill *Character*	Sets the character to be used as the kill, or stop, character. The value of the *Character* variable can be the octal representation of the character in the form `\NNN` where `\NNN` is a number or the character itself. For example, the character `\e` is `\145` in octal representation.

-noprepend Appends additional text after text already in the message body.
-norapid Displays text already in the message body. This is the default.
-prepend Appends additional text before text already in the message body. This is the default.
-rapid Does not display text already in the message body.

Profile Entries

`Msg-Protect:` Sets the protection level for your new message files.
`prompter-next:` Specifies the editor used after exiting the **prompter** command.

Files

$HOME/.mh_profile
Contains the MH user profile.
/tmp/prompter* Contains a temporary copy of a message.

Related Information

The **comp** command, **dist** command, **forw** command, **repl** command, **whatnow** command.

Mail Overview in *AIX 5L Version 5.2 System User's Guide: Communications and Networks*.

---

proto Command

Purpose

Constructs a prototype file for a file system.

Syntax

proto *Directory* [*Prefix*]

Description

The **proto** command creates a prototype file for a file system or part of a file system. The **mkfs** command uses the prototype file as input to construct a file system according to a predefined template. The prototype file consists of a recursive directory listing of every file on the file system, with its owner, group, and protection. It also contains the file from which the prototype file is to be initialized, formatted as described in the **mkfs** command.

Specify the root directory from which the prototype file is made with the *Directory* parameter. The prototype file includes the complete subtree below the *Directory* parameter, and is contained on the same file system as the base directory specified by the *Directory* parameter.

The *Prefix* parameter is added to the names of all the initialization files, forcing the initialization files to be taken from a place other than the prototype. Before the output from the **proto** command can be used with the **mkfs** command, the **mkfs** command needs a startup program, a file system size, and an i-node list size. Hard link information is not preserved with the **proto** command.

The collating sequence is determined by the **LANG** or **LC_COLLATE** environment variables.

Example

To make a prototype file for an existing file system `/works`, enter:

```
proto /works
```

If the /works file system contains two directories called dir1 and dir2, and the dir1 directory contains the file1 file, then the **proto** command displays:

```
#Prototype file for /works
d---  755  0 0
  dir1  d---  755  0 0
    file1         ----   644  0 0   /works/dir1/file1
        $
  dir2  d---  755  0 0
        $
   $
$
```

Files

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

Related Information

The **mkfs** command, **mkproto** command.

The File Systems in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices* explains file system types, management, structure, and maintenance.

The Files Overview in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices* provides information on working with files.

The National Language Support Overview in *AIX 5L Version 5.2 National Language Support Guide and Reference*.

Locales in *AIX 5L Version 5.2 National Language Support Guide and Reference*.

---

proxymngr Command

Purpose

Proxy manager service

Syntax

proxymngr [**-config** filename] [**-timeout** seconds] [**-retries** #] [**-verbose**]

Description

The **proxymngr** (proxy manager), is responsible for resolving requests from **xfindproxy** (and other similar clients), starting new proxies when appropriate, and keeping track of all of the available proxy services. The proxy manager strives to reuse existing proxies whenever possible.

There are two types of proxies that the proxy manager deals with, managed and unmanaged proxies.

A managed proxy is a proxy that is started on demand by the proxy manager.

An unmanaged proxy is started either at system boot time, or manually by a system administrator. The proxy manager is made aware of its existence, but no attempt is made by the proxy manager to start unmanaged proxies.

Flags

-config	Overrides the default **proxymngr** config file. See below for more details about the **proxymngr** config file.
-timeout	Sets the number of seconds between attempts made by the proxy manager to find an unmanaged proxy. The default is 10.
-retries	Sets the maximum number of retries made by the proxy manager to find an an unmanaged proxy. The default is 3.
-verbose	Causes various debugging and tracing records to be displayed as requests are received and proxies are started.

Proxy Manager Config File

The proxy manager maintains a local configuration file describing the proxy services available. This configuration file is installed in **/usr/X11R6.3/lib/X11/proxymngr/pmconfig** during the installation of **proxymngr**. The location of the configuration file can be overwritten using the **-config** command line flag.

Aside from lines starting with an exclamation point for comments, each line of the configuration file describes either an unmanaged or managed proxy service.

For unmanaged proxies, the format is:

<*service-name*> unmanaged <*proxy-address*>

service-name is the name of the unmanaged proxy service, and must not contain any spaces, for example `XFWP`. *service-name* is case insenstive.

proxy-address is the network address of the unmanaged proxy. The format of the address is specific to the *service-name*. For example, for the `XFWP` service, the *proxy-address* might be `firewall.x.org:100`.

If there is more than one entry in the config file with the same unmanaged *service-name*, the proxy manager will try to use the proxies in the order presented in the config file.

For managed proxies, the format is:

<*service-name*> managed <*command-to-start-proxy*>

service-name is the name of the managed proxy service, and must not contain any spaces, for example `LBX`. *service- name* is case insensitive.

command-to-start-proxy is the command executed by the proxy manager to start a new instance of the proxy. If *command- to-start-proxy* contains spaces, the complete command should be surrounded by single quotes. If desired, *command-to- start-proxy* can be used to start a proxy on a remote machine. The specifics of the remote execution method used to do this is not specified here.

Example: sample configuration file

```
! proxy manager config file
!
! Each line has the format:
!    <serviceName> managed <startCommand>
!        or
!    <serviceName> unmanaged <proxyAddress>
!
lbx managed /usr/X11R6.3/bin/lbxproxy
!
! substitute site-specific info
xfwp unmanaged firewall:4444
```

Proxy Manager Details

When the proxy manager gets a request from **xfindproxy** (or another similar client), its course of action will depend on the *service-name* in question.

For a managed proxy service, the proxy manager will find out if any of the already running proxies for this service can handle a new request. If not, the proxy manager will attempt to start up a new instance of the proxy (using the *command-to-start-proxy* found in the config file). If that fails, an error will be returned to the caller.

For an unmanaged proxy service, the proxy manager will look in the config file to find all unmanaged proxies for this service. If there is more than one entry in the config file with the same unmanaged *service-name*, the proxy manager will try to use the proxies in the order presented in the config file. If none of the unmanged proxies can satisfy the request, the proxy manager will timeout for a configurable amount of time (specified by **-timeout** or default of 10) and reattempt to find an unmanaged proxy willing to satisfy the request. The number of retries can be specified by the **-retries** argument, or a default of 3 will be used. If the retries fail, the proxy manager has no choice but to return an error to the caller (since the proxy manager can not start unmanaged proxy services).

---

prs Command (SCCS)

Purpose

Displays a Source Code Control System (SCCS) file.

Syntax

prs [**-a**] [**-d** *String*] [**-r** [*SID*] | [**-c** *Cutoff*]] [**-e** | **-l**] *File* ...

Description

The **prs** command first reads the specified files and then writes to standard output a part or all of a Source Code Control System (SCCS) file. If you specify a directory for the *File* parameter, the **prs** command performs the requested actions on all SCCS files (those with the **s.** prefix). If you specify a **-** (minus) for the *File* parameter, the **prs** command reads standard input and interprets each line as the name of an SCCS file. The **prs** command continues to read input until it reaches an end-of-file character.

Data Keywords

Data keywords specify the parts of an SCCS file to be retrieved and written to standard output. All parts of an SCCS file have an associated data keyword. There is no limit to the number of times a data keyword can be in a specified file.

The information that the **prs** command displays consists of user-supplied text and appropriate values (extracted from the SCCS file) substituted for the recognized data keywords in the order they are displayed in the specified file. The format of a data keyword value is either simple, in which the keyword substitution is direct, or multiline, in which the substitution is followed by a carriage return. Text consists of any characters other than recognized data keywords. Specify a tab character with **\t** (backslash, letter t) and a carriage return or new-line character with a **\n** (backslash, letter n). Remember to use the **\t** and **\n** with an extra **** (backslash) to prevent the shell from interpreting the **** and passing only the letter **t** or **n** to the **prs** command as text.

The following table lists the keywords associated with information in the delta table of the SCCS file:

Delta Table Keywords

Keyword	Data Represented	Value	Format
:R:	Release number	num	Simple
:L:	Level number	num	Simple

Delta Table Keywords

Keyword	Data Represented	Value	Format
:B:	Branch number	num	Simple
:S:	Sequence number	num	Simple
:I:	SCCS ID string (SID)	:R::L::B::S:	Simple
:Dy:	Year delta created	YY	Simple
:Dm:	Month delta created	MM	Simple
:Dd:	Day delta created	DD	Simple
:D:	Date delta created	YY/MM/DD	Simple
:Th:	Hour delta created	HH	Simple
:Tm:	Minute delta created	MM	Simple
:Ts:	Second delta created	SS	Simple
:T:	Time delta created	HH/MM/SS	Simple
:DT:	Delta type	D or R	Simple
:P:	User who created the delta	login name	Simple
:DS:	Delta sequence number	num	Simple
:DP:	Previous delta sequence number	num	Simple
:Dt:	Delta information	:DT::I::D::T::P::DS::DP:	Simple
:Dn:	Sequence numbers of deltas included	:DS: . . .	Simple
:Dx:	Sequence numbers of deltas excluded	:DS: . . .	Simple
:Dg:	Sequence numbers of deltas ignored	:DS: . . .	Simple
:Dl:	Sequence numbers of deltas included,excluded, and ignored	:Dn:/:Dx:/:Dg:	Simple
:Li:	Lines inserted by delta	num	Simple
:Ld:	Lines deleted by delta	num	Simple
:Lu:	Lines unchanged by delta	num	Simple
:DL:	Delta line statistics	:Li:/:Ld:/:Lu:	Simple
:MR:	MR numbers for delta	text	Multiline
:C:	Comments for delta	text	Multiline

The following table lists the keywords associated with header flags in the SCCS file:

Header Flag Keywords

Keyword	Data Represented	Value	Format
:Y:	Module type	text	Simple
:MF:	MR validation flag set	yes or no	Simple
:MP:	MR validation program name	text	Simple
:KF:	Keyword/error warning flag set	yes or no	Simple
:BF:	Branch flag set	yes or no	Simple

Header Flag Keywords

Keyword	Data Represented	Value	Format
:J:	Joint edit flag set	yes or no	Simple
:LK:	Locked releases	:R: . . .	Simple
:Q:	User-defined keyword	text	Simple
:M:	Module name	text	Simple
:FB:	Floor boundary	:R:	Simple
:CB:	Ceiling boundary	:R:	Simple
:Ds:	Default SID	:I:	Simple
:ND:	Null Delta flag set	yes or no	Simple
:FL:	Header flag list	text	Multiline

The following table lists the keywords associated with other parts of the SCCS file:

Other Keywords

Keyword	Data Represented	Value	Format
:UN:	User names	text	Multiline
:FD:	Descriptive text	text	Multiline
:BD:	Body of text	text	Multiline
:GB:	Text in a g-file	text	Multiline
:W:	A what string	:Z::M: \tab :I:	Simple
:A:	A what string	:Z::Y::M::I::Z:	Simple
:Z:	A what string delimiter	@(#)	Simple
:F:	SCCS file name	text	Simple
:PN:	SCCS file path name	text	Simple

Flags

Each flag or group of flags applies independently to each named file.

-a Writes information for the specified deltas, whether or not they have been removed (see the **rmdel** command). If you do not specify the **-a** flag, the **prs** command supplies information only for the specified deltas that have not been removed.

-c *Cutoff* Specifies a cutoff date and time for the **-e** and **-l** flags. Specify the *Cutoff* value in the following form:

```
YY[MM[DD[HH[MM[SS]]]]]
```

All omitted items default to their maximum values, so specifying -c8402 is the same as specifying -c840229235959. You can separate the fields with any non-numeric character. For example, you can specify `-c84/2/20,9:22:25` or `-c"84/2/20 9:22:25"` or `"-c84/2/20 9:22:25"`. The **-c** flag cannot be specified with the **-r** flag.

-d *String* Specifies the data items to be displayed. The string consists of optional text and SCCS file-data keywords. The string may include MBCS (multibyte character set) characters. If the string contains spaces, you must enclose the string in quotation marks.

-e Requests information for all deltas created earlier than and including the delta specified by the **-r** flag.

-l Requests information for all deltas created later than and including the delta specified by the **-r** flag.

-r [*SID*] Specifies the SCCS ID string (SID) of the delta for which the **prs** command will retrieve information. Do not enter a space between the **-r** flag and the optional SID parameter. If no SID is specified, the command retrieves the information for the SID of the highest numbered delta. The **-r** flag cannot be specified with the **-c** flag.

Exit Status

This command returns the following exit values:

0 Successful completion.
>0 An error occurred.

Examples

1. To display information on all deltas generated for SCCS file name **s.test.c** (including all deltas removed using the **rmdel** command), type:

   ```
   prs -a s.test.c
   ```

2. To display user login name, the number of lines inserted by delta, and the number of lines deleted by delta for SID `1.2` of `s.test.c`, type:

   ```
   prs -r1.2 -d":P:\n:Li:\n:Ld:" s.test.c
   ```

Files

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

Related Information

The **admin** command, **delta** command, **get** command, **sccshelp** command.

The **sccsfile** file format.

Source Code Control System (SCCS) Overview in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

---

prtacct Command

Purpose

Formats and displays files in **tacct** format.

Syntax

/usr/sbin/acct/prtacct [**-f** *Fields*] [**-v**] *File* [*"Heading"*]

Description

The **prtacct** command formats and displays any total-accounting file; these files are in **tacct** format. You can enter this command to view any **tacct** file, such as the daily reports on connect time, process time, disk usage, and printer usage. To specify a title for the report with the *Heading* parameter, enclose the heading text in *" "* (quotation marks).

Flags

-f *Fields* Selects fields to be displayed, using the field-selection mechanism of the **acctmerg** command.
-v Produces verbose output in which more precise notation is used for floating-point numbers.

Security

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

Examples

To format and display selected records from the total accounting file for connect-time accounting, you first must create a file upon which to execute the **prtacct** command. In this example, you create the **tacct** file using the **acctcon1** and **acctcon2** commands. Enter:

```
tail /var/adm/wtmp > wtmp.sav
```

```
acctcon1 -t < wtmp.sav | sort +1n +2 | acctcon2 > tacct
```

If you created this file previously to process connect-time accounting data, you do not need to create it again.

The next step uses the **prtacct** command with the **-f** flag to display the fields of data in the total-accounting file that you want to see. The text for a heading can be included in quotation marks. To view the login name, prime connect-time, and nonprime connect-time records, and include the heading, Connect-time Accounting, enter:

```
prtacct -f 2,11,12 tacct "Connect-time Accounting"
```

You can also use this command to format and display other total-accounting files, such as the daily reports on process time, disk usage, and printer usage.

Files

/usr/sbin/acct	The path to the accounting commands.
/var/adm/pacct	Current file for process accounting.
/var/adm/pacct*	Used if the **pacct** file gets too large.

Related Information

The **acctcon1** or **acctcon2** command, **acctdisk** or **acctdusg** command, **acctmerg** command, **acctprc1**, **acctprc2**, or **accton** command.

The **tacct** structure in the **acct** file format.

For more information about the Accounting System, the preparation of daily and monthly reports, and the accounting files, see the Accounting Overview in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices*.

Setting Up an Accounting System in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices* explains the steps you must take to establish an accounting system.

---

prtconf Command

Purpose

Displays system configuration information.

Syntax

prtconf [**-c**] [**-k**] [**-L**] [**-m**] [**-s**] [**-v**]

Description

If you run the **prtconf** command without any flags, it displays the system model, machine serial, processor type, number of processors, processor clock speed, cpu type, total memory size, network information, filesystem information, paging space information, and devices information.

Flags

-c	Displays cpu type, for example, 32-bit or 64-bit.
-k	Display the kernel in use, for example, 32-bit or 64-bit.
-L	Displays LPAR partition number and partition name if this is an LPAR partition, otherwise returns ″-1 NULL″.
-m	Displays system memory.
-s	Displays processor clock speed in MegaHertz.
-v	Displays the VPD found in the Customized VPD object class for devices.

Exit Status

0 The command completed successfully.

>0 An error occurred.

Examples

1. To display the system configuration information, enter:

   ```
   prtconf
   ```

   The system displays a message similar to the following:

   ```
   System Model: IBM,7025-F50
   Machine Serial Number: 1025778
   Processor Type: PowerPC_604
   Number Of Processors: 2
   Processor Clock Speed: 332 MHz
   CPU Type: 32-bit
   Kernel Type: 32-bit
   LPAR Info: -1 NULL
   Memory Size: 512 MB
   Good Memory Size: 512 MB
   Firmware Version: IBM,L02113
   Console Login: enable
   Auto Restart: false
   Full Core: false

   Network Information
   Host Name: vd01.austin.ibm.com
   IP Address: 9.3.207.112
   Sub Netmask: 255.255.255.128
   Gateway: 9.3.207.1
   Name Server: 9.3.199.2
   Domain Name: austin.ibm.com
   Paging Space Information


   Total Paging Space: 512MB
   Percent Used: 1%

   Volume Groups Information
   ==============================================================================
   rootvg:
   PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
   hdisk0            active            537         394         107..43..29..107..108
   ```

```
================================================================================

INSTALLED RESOURCE LIST

The following resources are installed on the machine.
+/- = Added or deleted from Resource List.
* = Diagnostic support not available.

Model Architecture: chrp
Model Implementation: Multiple Processor, PCI bus

+ sys0 00-00 System Object
+ sysplanar0 00-00 System Planar
+ mem0 00-00 Memory
etc.
```

2. To display the processor clock speed, enter:

   ```
   prtconf -s
   ```

   The system displays a message similar to the following:

   ```
   Processor Clock Speed: 332 MHz
   ```

3. To display the VPD for all physical devices in the Customized database, enter:

   ```
   prtconf -v
   ```

   The system displays a message similar to the following:

   ```
   INSTALLED RESOURCE LIST WITH VPD

   The following resources are installed on your machine.

     Model Architecture: chrp
     Model Implementation: Uni-Processor, PCI bus

     sys0             P1-C1       System Object
     sysplanar0                   System Planar
     mem0                         Memory
     L2cache0                     L2 Cache
     proc0            P1-C1       Processor

           Device Specific.(YL).....P1-C1

     pci0             P1          PCI Bus

           Device Specific.(YL).....P1

     isa0             P1          ISA Bus

           Device Specific.(YL).....P1

     fda0             P1/D1       Standard I/O Diskette Adapter

           Device Specific.(YL).....P1/D1

     fd0              P1-D1       Diskette Drive
     siokma0          P1/K1       Keyboard/Mouse Adapter

           Device Specific.(YL).....P1/K1

     sioka0           P1-K1       Keyboard Adapter
     kbd0             P1-K1-Lkbd  PS/2 keyboard
     sioma0           P1-O1       Mouse Adapter
     mouse0           P1-O1-Lmouse3 button mouse
     siota0           P1/Q1       Tablet Adapter

           Device Specific.(YL).....P1/Q1
   ```

```
paud0            P1/Q2       Ultimedia Integrated Audio

      Device Specific.(YL)........P1/Q2

ppa0             P1/R1       CHRP IEEE1284 (ECP) Parallel Port Adapter

      Device Specific.(YL)........P1/R1

sa0              P1/S1       Standard I/O Serial Port

      Device Specific.(YL)........P1/S1

tty0             P1/S1-L0    Asynchronous Terminal
sa1              P1/S2       Standard I/O Serial Port

      Device Specific.(YL)........P1/S2

ent0             P1/E1       IBM 10/100 Mbps Ethernet PCI Adapter (23100020)

      Network Address.............0004AC2A0419
      Displayable Message.........PCI Ethernet Adapter (23100020)
      Device Specific.(YL)........P1/E1

scsi0            P1/Z1       Wide/Fast-20 SCSI I/O Controller

      Device Specific.(YL)........P1/Z1

cd0              P1/Z1-A3    SCSI Multimedia CD-ROM Drive (650 MB)

      Manufacturer................IBM
      Machine Type and Model......CDRM00203
      ROS Level and ID............1_00
      Device Specific.(Z0)........058002028F000018
      Part Number.................97H7608
      EC Level....................F15213
      FRU Number..................97H7610

hdisk0           P1/Z1-A5    16 Bit SCSI Disk Drive (4500 MB)

      Manufacturer................IBM
      Machine Type and Model......DDRS-34560W
      FRU Number..................83H7105
      ROS Level and ID............53393847
      Serial Number...............RDHW5008
      EC Level....................F21433
      Part Number.................03L5256
      Device Specific.(Z0)........000002029F00003A
      Device Specific.(Z1)........00K0159S98G
      Device Specific.(Z2)........0933
      Device Specific.(Z3)........0299
      Device Specific.(Z4)........0001
      Device Specific.(Z5)........22
      Device Specific.(Z6)........F21390

bl0              P1.1-I2/G1  GXT255P Graphics Adapter

    GXT255P 2D Graphics Adapter:
      EC Level....................E76756
      FRU Number..................93H6267
      Manufacture ID..............IBM053
      Part Number.................93H6266
      Serial Number...............88074164
      Version.....................RS6K
      Displayable Message.........GXT255P
      ROM Level.(alterable).......02
      Product Specific.(DD).......00
      Product Specific.(DG).......00
```

```
        Device Specific.(YL)........P1.1-I2/G1

  pci1             P1.1        PCI Bus

        Device Specific.(YL)........P1.1
```

4. To display the kernel type in use:

   ```
   prtconf -k
   ```

   The system displays information for the kernel type as follows:

   ```
   Kernel Type: 32-bit
   ```

5. To display memory:

   ```
   prtconf -m
   ```

   The system displays memory, as follows:

   ```
      5.

   Memory Size: 512 MB
   ```

Files

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

Related Information

The **lsattr** command, **lsconn** command, **lsdev** command, **lsparent** command, **lscfg** command.

---

ps Command

Purpose

Shows current status of processes. This document describes the standard AIX **ps** command as well as the System V version of the **ps** command

Syntax

X/Open Standards

ps [**-A**] [**-N**] [**-a**] [**-d**] [**-e**] [**-f**] [**-k**] [**-l**] [**-F** *format*] [**-o** *Format*] [**-c** *Clist*] [**-G** *Glist*] [**-g** *Glist*] [**-m**] [**-n** *NameList*] [**-p** *Plist*] [**-t** *Tlist*] [**-U** *Ulist*] [**-u** *Ulist*]

Berkeley Standards

ps [**a**] [**c**] [**e**] [**ew**] [**eww**] [**g**] [**n**] [**U**] [**w**] [**x**] [**l** | **s** | **u** | **v**] [**t** *Tty*] [*ProcessNumber*]

Description

The **ps** command writes the current status of active processes and (if the **-m** flag is given) associated kernel threads to standard output. Note that while the **-m** flag displays threads associated with processes using extra lines, you must use the **-o** flag with the **THREAD** field specifier to display extra thread-related columns.

Without flags, the **ps** command displays information about the current workstation. The **-f**, **-o**, **l**, **-l**, **s**, **u**, and **v** flags only determine how much information is provided about a process; they do not determine which processes are listed. The **l**, **s**, **u**, and **v** flags are mutually exclusive.

With the **-o** flag, the **ps** command examines memory or the paging area and determines what the command name and parameters were when the process was created. If the **ps** command cannot find this information, the command name stored in the kernel is displayed in square brackets.

The **COLUMNS** environment variable overrides the system-selected, horizontal screen size.

The command-line flags that accept a list of parameters (the **-o**, **-G**, **-g**, **-p**, **-t**, **-U**, and **-u** flags) are limited to 128 items. For example, the **-u** *Ulist* flag can specify no more than 128 users.

Depending on the flags used with the **ps** command, column headings are displayed above the information displayed to standard output. The headings are defined in the following list (flags that cause these headings to be displayed are shown in parentheses):

ADDR (**-l** and **l** flags) Contains the segment number of the process stack, if normal; if a kernel process, the address of the preprocess data area.

BND (**-o THREAD** flag) The logical processor number of the processor to which the kernel thread is bound (if any). For a process, this field is shown if all its threads are bound to the same processor.

C (**-f**, **l**, and **-l** flags) CPU utilization of process or thread, incremented each time the system clock ticks and the process or thread is found to be running. The value is decayed by the scheduler by dividing it by 2 once per second. For the `sched_other` policy, CPU utilization is used in determining process scheduling priority. Large values indicate a CPU intensive process and result in lower process priority whereas small values indicate an I/O intensive process and result in a more favorable priority.

CMD (**-f**, **-l**, and **l** flags) Contains the command name. The full command name and its parameters are displayed with the **-f** flag.

COMMAND
(**s**, **u**, and **v**) Contains the command name. The full command name and its parameters are displayed with the **-f** flag.

F Field Table

Flags	Hexadecimal Value	Definition
SLOAD	0x00000001	Indicates that the process is operating in core memory.
SNOSWAP	0x00000002	Indicates that the process cannot be swapped out.
STRC	0x00000008	Indicates that the process is being traced.
SWTED	0x00000010	Indicates that the process stopped while being traced.
SFWTED	0x00000020	Indicates that the process stopped after a call to the **fork** subroutine, while being traced.
SEWTED	0x00000040	Indicates that the process stopped after a call to the **exec** subroutine, while being traced.
SLWTED	0x00000080	Indicates that the process stopped after a call to the **load** or **unload** subroutine, while being traced.
SFIXPRI	0x00000100	Indicates that the process has a fixed priority, ignoring the **pcpu** field descriptor.
SKPROC	0x00000200	Indicates a Kernel process.
SOMASK	0x00000400	Indicates restoration of the old mask after a signal is received.
SWAKEONSIG	0x00000800	Indicates that the signal will abort the **sleep** subroutine. The contents must *not* be equal to those of the **PCATCH** flag. The contents of both **PCATCH** and **SWAKEONSIG** must be greater than those of **PMASK**.
SUSER	0x00001000	Indicates that the process is in user mode.
SLKDONE	0x00002000	Indicates that the process has done locks.

F Field Table

Flags	Hexadecimal Value	Definition
STRACING	0x00004000	Indicates that the process is a debugging process.
SMPTRACE	0x00008000	Indicates multi-process debugging.
SEXIT	0x00010000	Indicates that the process is exiting.
SSEL	0x00020000	Indicates that the processor is selecting: wakeup/waiting danger.
SORPHANPGRP	0x00040000	Indicates an orphaned process group.
SNOCNTLPROC	0x00080000	Indicates that the session leader relinquished the controlling terminal.
SPPNOCLDSTOP	0x00100000	Indicates that the **SIGHLD** signal is *not* sent to the parent process when a child stops.
SEXECED	0x00200000	Indicates that process has been run.
SJOBSESS	0x00400000	Indicates that job control was used in the current session.
SJOBOFF	0x00800000	Indicates that the process is free from job control.
PSIGDELIVERY	0x01000000	Indicates that the process is used by the program-check handler.
SRMSHM	0x02000000	Indicates that the process removed shared memory during a call to the **exit** subroutine.
SSLOTFREE	0x04000000	Indicates that the process slot is free.
SNOMSG	0x08000000	Indicates that there are no more **uprintf** subroutine messages.

F (**-l** and **l** flags) Some of the more important F field flags (hexadecimal and additive) associated with processes and threads are shown in the following table:

F Field Table

Flags	Hexadecimal Value	Definition
SLOAD	0x00000001	Indicates that the process is operating in core memory.
SNOSWAP	0x00000002	Indicates that the process cannot be swapped out.
STRC	0x00000008	Indicates that the process is being traced.
SKPROC	0x00000200	Indicates a kernel process.
SEXIT	0x00010000	Indicates that the process is exiting.
SEXECED	0x00200000	Indicates that the process has been run.
SEXECING	0x01000000	Indicates that the process is execing (performing an exec).
TKTHREAD	0x00001000	Indicates that the thread is a kernel-only thread.

Note: You can see the definitions of all process and thread flags by consulting the **p_flags** and **t_flags** fields in the **/usr/include/sys/proc.h** and **/usr/include/sys/thread.h** files respectively.

LIM (**v** flag) The soft limit on memory used, specified via a call to the **setrlimit** subroutine. If no limit has been specified, then shown as `xx`. If the limit is set to the system limit, (unlimited), a value of `UNLIM` is displayed.

NI (**-l** and **l** flags) The nice value; used in calculating priority for the `sched other` policy.

PID (all flags) The process ID of the process.

PGIN (**v** flag) The number of disk I/Os resulting from references by the process to pages not loaded in core.

PPID (**-f**, **l**, and **-l** flags) The process ID of the parent process.

PRI (**-l** and **l** flags) The priority of the process or kernel thread; higher numbers mean lower priority.

RSS (**v** flag) The real-memory (resident set) size of the process (in 1KB units).

S (**-l** and **l** flags) The state of the process or kernel thread :

For processes:

O Nonexistent

A Active

W Swapped

I Idle (waiting for startup)

Z Canceled

T Stopped

For kernel threads:

O Nonexistent

R Running

S Sleeping

W Swapped

Z Canceled

T Stopped

SC (**-o THREAD** flag) The suspend count of the process or kernel thread. For a process, the suspend count is defined as the sum of the kernel threads suspend counts.

SCH (**-o THREAD** flag) The scheduling policy for a kernel thread. The policies `sched_other`, `sched_fifo`, and `sched_rr` are respectively displayed using: `0`, `1`, `2`.

SIZE (**v** flag) The virtual size of the data section of the process (in 1KB units).

SSIZ (**s** flag) The size of the kernel stack. This value is always 0 (zero) for a multi-threaded process.

STAT (**s**, **u**, and **v** flags) Contains the state of the process:

0 Nonexistent

A Active

I Intermediate

Z Canceled

T Stopped

K Available kernel process

STIME

(**-f** and **u** flags) The starting time of the process. The **LANG** environment variables control the appearance of this field.

SZ (**-l** and **l** flags) The size in 1KB units of the core image of the process.

THCNT
(**-o thcount** flag) The number of kernel threads owned by the process.

TID (**-o THREAD** flag) The thread ID of the kernel thread.

TIME (all flags) The total execution time for the process.

TRS (**v** flag) The size of resident-set (real memory) of text.

TSIZ (**v** flag) The size of text (shared-program) image.

TTY (all flags) The controlling workstation for the process:

- The process is not associated with a workstation.

? Unknown.

Number
The TTY number. For example, the entry 2 indicates TTY2.

UID (**-f**, **-l**, and **l** flags) The user ID of the process owner. The login name is printed under the **-f** flag.

USER (**u** flag) The login name of the process owner.

WCHAN
(**-l** flag) The event for which the process or kernel thread is waiting or sleeping. For a kernel thread, this field is blank if the kernel thread is running. For a process, the wait channel is defined as the wait channel of the sleeping kernel thread if only one kernel thread is sleeping; otherwise a star is displayed.

WCHAN
(**l** flag) The event on which process is waiting (an address in the system). A symbol is chosen that classifies the address, unless numerical output is requested.

%CPU (**u** and **v** flags) The percentage of time the process has used the CPU since the process started. The value is computed by dividing the time the process uses the CPU by the elapsed time of the process. In a multi-processor environment, the value is further divided by the number of available CPUs because several threads in the same process can run on different CPUs at the same time. (Because the time base over which this data is computed varies, the sum of all **%CPU** fields can exceed 100%.)

%MEM
(**u** and **v** flags) The percentage of real memory used by this process.

A process that has exited and has a parent that has not yet waited for the process is marked <defunct>. A process that is blocked trying to exit is marked <exiting>. The **ps** command attempts to determine the file name and arguments given when the process was created by memory or by the swap area.

Notes:

1. The process can change while the **ps** command is running. Some data displayed for defunct processes is irrelevant.
2. The **ps** program examines memory to retrieve the file name and arguments used when the process was created. However, a process can destroy information, making this method of retrieving file name and arguments unreliable.

Flags

The following flags are preceded by a **-** (minus sign):

-A Writes to standard output information about all processes.

-a	Writes to standard output information about all processes, except the session leaders and processes not associated with a terminal.
-c *Clist*	Displays only information about processes assigned to the workload management classes listed in the *Clist* variable. The *Clist* variable is either a comma separated list of class names or a list of class names enclosed in double quotation marks (″ ″), which is separated from one another by a comma or by one or more spaces, or both.
-d	Writes information to standard output about all processes, except the session leaders.
-e	Writes information to standard output about all processes, except kernel processes.
-F *Format*	Same as **-o** *Format*
-f	Generates a full listing.
-G *Glist*	Writes information to standard output only about processes that are in the process groups listed for the *Glist* variable. The *Glist* variable is either a comma-separated list of process group identifiers or a list of process group identifiers enclosed in double quotation marks (″ ″) and separated from one another by a comma or by one or more spaces. This flag is equivalent to the **-g** *Glist* flag.
-g *Glist*	This flag is equivalent to the **-G** *Glist* flag.
-k	Lists kernel processes.
-l	Generates a long listing. See also the **l** flag.
-m	Lists kernel threads as well as processes. Output lines for processes are followed by an additional output line for each kernel thread. This flag does not display thread-specific fields (**bnd**, **scount**, **sched**, **thcount**, and **tid**), unless the appropriate **-o** *Format* flag is specified.
-N	Gathers no thread statistics. With this flag, **ps** reports those statistics that can be obtained by not traversing through the threads chain for the process.
-n *NameList*	Specifies an alternative system name-list file in place of the default. The operating system does not use the **-n** flag because information is supplied directly to the kernel.

-o *Format*

Displays information in the format specified by the *Format* variable. Multiple field specifiers can be specified for the *Format* variable. The *Format* variable is either a comma-separated list of field specifiers or a list of field specifiers enclosed within a set of ″ ″ (double-quotation marks) and separated from one another by a comma or by one or more spaces, or both.

Each field specifier has a default header. The default header can be overridden by appending an **=** (equal sign) followed by the user-defined text for the header. The fields are written in the order specified on the command line in column format. The field widths are specified by the system to be at least as wide as the default or user-defined header text. If the header text is null, (such as if **-o user=** is specified), the field width is at least as wide as the default header text. If all header fields are null, no header line is written.

The following field specifiers are recognized by the system:

args Indicates the full command name being executed. All command-line arguments are included, though truncation may occur. The default header for this field is **COMMAND**.

bnd Indicates to which (if any) processor a process or kernel thread is bound. The default header for this field is **BND**.

class Indicates the workload management class assigned to the process. the default header for this field is **CLASS**.

comm Indicates the short name of the command being executed. Command-line arguments are not included. The default header for this field is **COMMAND**.

cpu Determines process scheduling priority. CPU utilization of process or thread, incremented each time the system clock ticks and the process or thread is found to be running. The value is decayed by the scheduler by dividing it by 2 once per second. For the sched_other policy, Large values indicate a CPU intensive process and result in lower process priority whereas small values indicate an I/O intensive process and result in a more favorable priority.

etime Indicates the elapsed time since the process started. The elapsed time is displayed in the following format:

[[*dd***-**]*hh***:**]*mm***:***ss*

where *dd* specifies the number of days, *hh* specifies the number of hours, *mm* specifies the number of minutes, and *ss* specifies the number of seconds. The default header for this field is **ELAPSED**.

group Indicates the effective group ID of the process. The textual group ID is displayed. If the textual group ID cannot be obtained, a decimal representation is used. The default header for this field is **GROUP**.

nice Indicates the decimal value of the process nice value. The default header for this field is **NI**.

-o *Format (Continued)*

pcpu Indicates the ratio of CPU time used to CPU time available, expressed as a percentage. The default header for this field is **%CPU**.

pgid Indicates the decimal value of the process group ID. The default header for this field is **PGID**.

pid Indicates the decimal value of the process ID. The default header for this field is **PID**.

ppid Indicates the decimal value of the parent process ID. The default header for this field is **PPID**.

rgroup Indicates the real group ID of the process. The textual group ID is displayed. If the textual group ID cannot be obtained, a decimal representation is used. The default header for this field is **RGROUP**.

ruser Indicates the real user ID of the process. The textual user ID is displayed. If the textual user ID cannot be obtained, a decimal representation is used. The default header for this field is **RUSER**.

scount Indicates the suspend count for a kernel thread. The default header for this field is **SC**.

sched Indicates the scheduling policy for a kernel thread. The default header for this field is **SCH**.

tag Indicates the Workload Manager application tag. The default header for this field is **TAG**. The tag is a character string up to 30 characters long and may be truncated when displayed by **ps**. For processes that do not set their tag, this field displays as a **-** (hyphen).

thcount
Indicates the number of kernel threads owned by the process. The default header for this field is **THCNT**.

-o *Format (Continued)*

THREAD

Indicates the following fields:

- User name (the **uname** field)
- Process and parent process IDs for processes (the **pid** and **ppid** fields)
- Kernel thread ID for threads (the **tid** field)
- The state of the process or kernel thread (the **S** field)
- The CPU utilization of the process or kernel thread (the **C** field)
- The priority of the process or kernel thread (the **PRI** field)
- The suspend count of the process or kernel thread (the **scount** field)
- The wait channel of the process or kernel thread (the **WCHAN** field)
- The flags of the process or kernel thread (the **F** field)
- The controlling terminal of the process (the **tty** field)
- The CPU to which the process or kernel thread is bound (the **bnd** field)
- The command being executed by the process (the **comm** field).

Threads are not actually displayed with the **-o THREAD** flag, unless the **-m** flag is also specified.

tid Indicates the thread ID of a kernel thread. The default header for this field is **TID**.

time Indicates the cumulative CPU time since the process started. The time is displayed in the following format:

[*dd***-**]*hh***:***mm***:***ss*

where *dd* specifies the number of days, *hh* specifies the number of hours, *mm* specifies the number of minutes, and *ss* specifies the number of seconds. The default header for this field is **TIME**.

tty Indicates the controlling terminal name of the process. The default header for this field is **TT**.

user Indicates the effective user ID of the process. The textual user ID is displayed. If the textual user ID cannot be obtained, a decimal representation is used. The default header for this field is **USER**.

vsz Indicates, as a decimal integer, the size in kilobytes of the process in virtual memory. The default header for this field is **VSZ**.

-o *Format (Continued)* Otherwise, multiple fields in a specified format can be displayed by the *Format* variable, including field descriptors. If field descriptors are used in the *Format* variable, it must be enclosed in double quotation marks (″ ″). The following table shows how field descriptors correspond to field specifiers:

Field Descriptors	Field Specifiers	Default Headers
%a	args	COMMAND
%c	comm	COMMAND
%t	etime	ELAPSED
%G	group	GROUP
%n	nice	NI
%C	pcpu	%CPU
%r	pgid	PGID
%p	pid	PID
%P	ppid	PPID
%g	rgroup	RGROUP
%u	ruser	RUSER
%x	time	TIME
%y	tty	TTY
%U	user	USER
%z	vsz	VSZ

-p *Plist* Displays only information about processes with the process numbers specified for the *Plist* variable. The *Plist* variable is either a comma separated list of process ID numbers or a list of process ID numbers enclosed in double quotation marks (″ ″) and separated from one another by a comma or by one or more spaces, or both.

-t *Tlist* Displays only information about processes associated with the workstations listed in the *Tlist* variable. The *Tlist* variable is either a comma separated list of workstation identifiers or a list of workstation identifiers enclosed in double quotation marks (″ ″) and separated from one another by a comma or by one or more spaces, or both.

-U *Ulist* Displays only information about processes with the user ID numbers or login names specified for the *Ulist* variable. The *Ulist* variable is either a comma-separated list of user IDs or a list of user IDs enclosed in double quotation marks (″ ″) and separated from one another by a comma and one or more spaces. In the listing, the **ps** command displays the numerical user ID unless the **-f** flag is used; then the command displays the login name. This flag is equivalent to the **-u** *Ulist* flag. See also the **u** flag.

-u *Ulist* This flag is equivalent to the **-U** *Ulist* flag.

Options

The following options are not preceded by a **-** (minus sign):

a Displays information about all processes with terminals (ordinarily only the user's own processes are displayed).

c Displays the command name, as stored internally in the system for purposes of accounting, rather than the command parameters, which are kept in the process address space.

e Displays the environment as well as the parameters to the command, up to a limit of 80 characters.

ew Wraps display from the **e** flag one extra line.

eww Wraps display from the **e** flag as many times as necessary.

g Displays all processes.

l Displays a long listing having the **F**, **S**, **UID**, **PID**, **PPID**, **C**, **PRI**, **NI**, **ADDR**, **SZ**, **PSS**, **WCHAN**, **TTY**, **TIME**, and **CMD** fields.

n Displays numerical output. In a long listing, the **WCHAN** field is printed numerically rather than symbolically. In a user listing, the **USER** field is replaced by a **UID** field.

s Displays the size (SSIZ) of the kernel stack of each process (for use by system maintainers) in the basic output format. This value is always 0 (zero) for a multi-threaded process.

t *Tty* Displays processes whose controlling tty is the value of the *Tty* variable, which should be specified as printed by the **ps** command; that is, 0 for terminal **/dev/tty/0**, lft0 for **/dev/lft0** , and pts/2 for **/dev/pts/2**.

u Displays user-oriented output. This includes the **USER**, **PID**, **%CPU**, **%MEM**, **SZ**, **RSS**, **TTY**, **STAT**, **STIME**, **TIME**, and **COMMAND** fields.

v Displays the **PGIN**, **SIZE**, **RSS**, **LIM**, **TSIZ**, **TRS**, **%CPU**, **%MEM** fields.

w Specifies a wide-column format for output (132 columns rather than 80). If repeated, (for example, `ww`), uses arbitrarily wide output. This information is used to decide how much of long commands to print.

x Displays processes without a controlling terminal in addition to processes with a controlling terminal.

Exit Status

This command returns the following exit values:

0 Successful completion.

>0 An error occurred.

Examples

1. To display all processes, type:

   ```
   ps -e -f
   ```

2. To list processes owned by specific users, type:

   ```
   ps -f -l -ujim,jane,su
   ```

3. To list processes that are associated with the **/dev/console** and **/dev/tty1** ttys, type:

   ```
   ps -t console,tty/1
   ```

4. To list processes not associated with a terminal, type:

   ```
   ps -t -
   ```

5. To display a specified format with field specifiers, type:

   ```
   ps -o ruser,pid,ppid=parent,args
   ```

   The output is:

   ```
   RUSER   PID     parent  COMMAND
   helene  34      12      ps -o ruser,pid,ppid=parent,args
   ```

6. To display a specified format with field descriptors, type:

   ```
   ps -o "< %u > %p %y : %a"
   ```

   The output is:

   ```
   < RUSER  >      PID     TT :    COMMAND
   < helene >      34      pts/3 : ps -o < %u > %p %y : %a
   ```

7. To display information about processes and kernel threads controlled by the current terminal, type:

   ```
   ps -lm
   ```

   The output is similar to:

   ```
        F S UID  PID PPID  C PRI NI ADDR  SZ WCHAN   TTY  TIME  CMD
   240003 A  26 8984 7190  1  60 20 2974 312       pts/1  0:00  -ksh
      400 S   -    -    -  1  60  -    -   -           -     -  -
   200005 A  26 9256 8984 15  67 20 18ed 164       pts/1  0:00  ps
        0 R   -    -    - 15  67  -    -   -           -     -  -
   ```

8. To display information about all processes and kernel threads, type:

   ```
   ps -emo THREAD
   ```

   The output is similar to:

   ```
   USER   PID  PPID  TID S  C PRI SC   WCHAN   FLAG   TTY BND  CMD
   jane  1716 19292    - A 10  60  1       * 260801 pts/7   -  biod
      -     -     - 4863 S  0  60  0 599e9d8   8400     -   -  -
      -     -     - 5537 R 10  60  1 5999e18   2420     -   3  -
   luke 19292 18524    - A  0  60  0 586ad84 200001 pts/7   -  -ksh
   ```

```
   -     -     - 7617 S  0  60  0 586ad84    400     -   -  -
luke 25864 31168    - A 11  65  0       - 200001 pts/7   -  -
   -     -     - 8993 R 11  65  0       -      0     -   -  -
```

Files

/**usr**/**bin**/**ps**	Contains the **ps** command.
/etc/passwd	Specifies user ID information.
/dev/pty*	Indicates workstation (**PTY**) names.
/dev/tty*	Indicates workstation (**TTY**) names.

Related Information

The **kill** command, **nice** command.

Using ps to Identify CPU-Intensive Programs in *AIX 5L Version 5.2 Performance Management Guide*.

System V ps command

Syntax

/usr/sysv/bin/ps [**-a**] [**-A**] [**-c**] [**-d**] [**-e**] [**-f**] [**-j**] [**-l**] [**-L**] [**-P**] [**-y**] [**-g** *pgrplist*] [**-o** *format*] [**-p** *proclist*] [**-s** *sidlist*] [**-t** *termlist*] [{ **-u** | **-U** } *uidlist*] [**-G** *grplist*]

Description

The **ps** command prints information about active processes. Without flags, **ps** prints information about processes associated with the controlling terminal. The output contains the process ID, terminal identifier, cumulative execution time, and the command name. The information displayed with flags varies accordingly.

Output

Depending on the flags used with the **ps** command, column headings vary for the information displayed . The headings are defined in the following list (flags that cause these headings to appear are shown in parentheses):

F (-l) Flags (hexadecimal and additive) associated with the process, or the thread if the **-L** option is specified. Some of the more important F field flags (hexadecimal and additive) associated with processes and threads are shown below:

F Field Table

Flags	Hexadecimal Value	Definition
SLOAD	0x00000001	Indicates that the process is operating in core memory.
SNOSWAP	0x00000002	Indicates that the process cannot be swapped out.
STRC	0x00000008	Indicates that the process is being traced.
SKPROC	0x00000200	Indicates a Kernel process.
SEXIT	0x00010000	Indicates that the process is exiting.
SEXECED	0x00200000	Indicates that process has been run.
SEXECING	0x01000000	Indicates that the process is execing (performing an exec).
TKTHREAD	0x00001000	Indicates that the thread is a kernel only thread.

Note: You can see the definitions of all process and thread flags by referring to the **p_flags** and **t_flags** fields in the **/usr/include/sys/proc.h** and **/usr/include/sys/thread.h** files respectively.

S (-l) The state of the process or kernel thread :

For processes:

O Nonexistent

A Active

W Swapped

I Idle

Z Canceled

T Stopped

For kernel threads:

O Nonexistent

R Running

S Sleeping

W Swapped

Z Canceled

T Stopped

UID (-f,- l)
The user ID number of the process (the login name is printed under the -f option).

PID (all)
The process ID of the process.

PPID (-f,-l)
The process ID of the parent process.

CLS (-c)
Scheduling class for the process . Printed only when the **-c** flag is used.

NI (-l) The nice value of the process used in calculating priority for the **sched_other** policy.

PRI (-c, -l)
The priority of the process or kernel thread. Higher numbers mean lower priority.

ADDR (-l)
Contains the segment number of the process stack, if normal; if a kernel process, the address of the preprocess data area.

SZ (-l) The size in pages of the core image of the process.

WCHAN(-l)
The event for which the process or kernel thread is waiting or sleeping. For a kernel thread, this field is blank if the kernel thread is running. For a process, the wait channel is defined as the wait channel of the sleeping kernel thread if only one kernel thread is sleeping; otherwise a star is displayed.

STIME (-f,-u)
The starting time of the process. The **LANG** environment variables control the appearance of this field.

TTY (all)
The controlling workstation for the process:

- The process is not associated with a workstation.

? Unknown

TIME (all)
The total execution time for the process.

LTIME (-L)
The execution time for an individual LWP.

CMD (all)
Contains the command name. The full command name and its parameters are displayed with the **-f** flag.

LWP (-L)
The tid of the kernel thread.

NLWP(-Lf)
The number of kernel threads in the process.

PSR (-P)
The logical processor number of the processor to which the kernel thread is bound (if any). For a process, this field is shown if all its threads are bound to the same processor.

RSS (-ly)
The real memory (resident set) size of the process in pages.

Format

The following list describes the field specifiers recognized by the system. These field specifiers can be used with the **-o** flag to specify the format for the output of the **ps** command.

The field specifiers recognized by the system are:

addr Indicates the segment number of the process stack, if normal; if a kernel process, the address of the preprocess data area. The default header for this field is **ADDR**.

args Indicates the full command name being executed. All command line arguments are included, though truncation may occur. The default header for this field is **COMMAND**.

c CPU utilization of process or thread, incremented each time the system clock ticks and the process or thread is found to be running. The value is decayed by the scheduler by dividing it by 2 once per second. For the **sched_other** policy, CPU utilization is used in determining process scheduling priority. Large values indicate a CPU intensive process and result in lower process priority whereas small values indicate an I/O intensive process and result in a more favorable priority. The default header for this field is **C**.

class Indicates the scheduling policy for a kernel thread. The policies are sched_other , sched_fifo and sched_rr. The default header for this field is **CLS**.

comm Indicates the short name of the command being executed. Command line arguments are not included. The default header for this field is **COMMAND**.

etime Indicates the elapsed time since the process started. The elapsed time is displayed in the format
`[[ dd -] hh: ]mm :ss`

where *dd* specifies the number of days, *hh* specifies the number of hours, *mm* specifies the number of minutes, and *ss* specifies the number of seconds.

The default header for this field is **ELAPSED**.

f Indicates flags (hexadecimal and additive) associated with the process. The default header for this field is **COMMAND**.

fname Indicates the first 8 bytes of the base name of the process's executable file. The default header for this field is **COMMAND**.

gid Indicates the effective group ID number of the process as a decimal integer. The default header for this field is **GID**.

group Indicates the effective group ID of the process. The textual group ID is displayed. If the textual group ID cannot be obtained, a decimal representation is used. The default header for this field is **GROUP**.

lwp Indicates the thread ID of the kernel thread. The default header for this field is **TID**.

nice Indicates the decimal value of the process nice value. The default header for this field is **NI**.

nlwp Indicates the number of kernel threads owned by the process. The default header for this field is **THCNT**.

pcpu Indicates the ratio of CPU time used to CPU time available, expressed as a percentage. The default header for this field is **%CPU**.

pgid Indicates the decimal value of the process group ID. The default header for this field is **PGID**.

pid Indicates the decimal value of the process ID. The default header for this field is **PID**.

pmem Indicates the percentage of real memory used by this process. The default header for this field is **%MEM**.

ppid Indicates the decimal value of the parent process ID. The default header for this field is **PPID**.

pri Indicates the priority of the process or kernel thread ; higher numbers mean lower priority. The default header for this field is **PRI**.

psr Indicates the logical processor number of the processor to which the kernel thread is bound (if any). The default header for this field is **PSR**.

rgid Indicates the real group ID number of the process as a decimal integer. The default header for this field is **RGID**.

rgroup
Indicates the real group ID of the process. The textual group ID is displayed. If the textual group ID cannot be obtained, a decimal representation is used. The default header for this field is **RGROUP**.

rss Indicates the real memory (resident set) size of the process (in 1 KB units). The default header for this field is **RSS**.

ruid Indicates the real user ID number of the process as a decimal integer. The default header for this field is **RUID**.

ruser Indicates the real user ID of the process. The textual user ID is displayed. If the textual user ID cannot be obtained, a decimal representation is used. The default header for this field is **RUSER**.

s Indicates the state of the process. The default header for this field is **S**.

sid Indicates the process ID of the session leader. The default header for this field is **SID**.

stime Indicates the starting time of the process. The LANG environment variables control the appearance of this field. The default header for this field is **STIME**.

time Indicates the cumulative CPU time since the process started. The time is displayed in the same format as in **etime**. The default header for this field is **TIME**.

tty Indicates the controlling terminal name of the process. The default header for this field is **TT**.

uid Indicates the effective user ID number of the process as a decimal integer. The default header for this field is **UID**.

user Indicates the effective user ID of the process. The textual user ID is displayed. If the textual user ID cannot be obtained, a decimal representation is used. The default header for this field is **USER**.

vsz Indicates, as a decimal integer, the size in kilobytes of the core image of the process. The default header for this field is **VSZ**.

wchan
Indicates the event for which the process or kernel thread is waiting or sleeping. For a kernel thread, this field is blank if the kernel thread is running. For a process, the wait channel is defined as the wait channel of the sleeping kernel thread if only one kernel thread is sleeping; otherwise a star is displayed.

The default header for this field is **WCHAN**.

Flags

Some flags accept lists as arguments. Items in a list can be either separated by commas or else enclosed in double quotes and separated by commas or spaces. Values for *proclist* and *pgrplist* must be numeric.

-a	Writes to standard output information about all processes, except the session leaders and processes not associated with a terminal.
-A	Writes to standard output information about all processes.
-c	Prints information in a format that reflects scheduler properties. The **-c** flag affects the output of the **-f** and **-l** flags, as described below.
-d	Writes to standard output information about all processes, except the session leaders.
-e	Writes to standard output information about all processes, except kernel processes.
-f	Generates a full listing.
-g *pgrplist*	Writes to standard output information only about processes that are in the process groups specified by *pgrplist*. Values for *pgrplist* must be numeric.
-G *grplist*	Writes to standard output information only about processes that are in the process groups specified by *grplist*. The **-G** flag accepts group names.
-j	Displays session ID and process group ID.
-l	Generates a long listing.
-L	Prints status of active threads within a process.
-o *format*	Displays information in the format specified by *format*. Multiple field specifiers can be specified for the format variable. The field specifiers that can be used with the **-o** flag are described above in the Format section.
-p *proclist*	Displays information only about processes with the process numbers specified by *proclist*. Values for *proclist* must be numeric.
-P	Displays the logical processor number of the processor to which the primary kernel thread of the process is bound (if any).
-s *sidlist*	Displays all processes whose session leader's IDs are specified by *sidlist*.
-t *termlist*	Displays information only about processes associated with the terminals specified by *termlist*.
-u *uidlist*	Displays information only about processes with the user ID numbers or login names specified by *uidlist*.
-U *uidlist*	Displays information only about processes with the user ID numbers or login names specified by *uidlist*.
-y	When combined with the **-l** option, changes the long listing so that it prints the ″RSS″ and ″SZ″ fields in kilobytes and does not print the ″F″ and ″ADDR″ fields.

Exit Status

This command returns the following exit values:

0 Successful completion.

>0 An error occurred.

Examples

1. To display all processes, enter:
   ```
   ps -e -f
   ```
2. To list processes owned by the user 'guest', enter:
   ```
   ps -f -l -u guest
   ```
3. To list processes that are associated with the **/dev/pts/0** and **/dev/pts/1** terminals, enter:
   ```
   ps -t pts/0,pts/1
   ```
4. To list processes not associated with a terminal, enter:
   ```
   ps -t -
   ```
5. To display a specified format with field specifiers, enter:
   ```
   ps -o ruser,pid,ppid,args
   ```
6. To display information about all kernel threads in a process, enter:
   ```
   ps -L
   ```
7. To display session id and proces group ids of all the processes, enter:
   ```
   ps -jA
   ```
8. To display the scheduling class and priority of processes, enter:
   ```
   ps -c -l
   ```
9. To display information about kernel threads and the number of kernel threads in a process, enter:
   ```
   ps -L -f
   ```
10. To display the processor to which the process or kernel thread is bound to, enter:
    ```
    ps  -P
    ```

Files

/usr/sysv/bin/ps	Contains the System V R4 **ps** command.
/etc/passwd	Contains the user ID information.
/dev/pty*	Indicates workstation (PTY) names.
/dev/tty*	Indicates workstation (TTY) names.

Related Information

The **kill** command, **nice** command.

---

ps4014 Command

Purpose

Converts a Tektronix 4014 file to PostScript format.

Syntax

ps4014 [**-m**] [**-C**] [**-N**] [**-R**] [**-s***Width*,*Height*] [**-l***Left*,*Bottom*] [**-S***Width*] [**-p***OutFile*] [*File*]

Description

The **ps4014** command reads in a Tektronix 4014 format file and converts it to PostScript format for printing on a PostScript printer. If no file is specified, the standard input is used. The resulting PostScript file can be directed to standard output or to a named file.

Note: By default, the 4014 image is scaled to occupy nearly the entire page in a landscape orientation.

Flags

Note: The **-m**, **-C**, and **-N** flags specify values for 4014 hardware options that affect the interpretation of 4014 commands.

-C	Causes a carriage return to move the pen position to the left margin but not down to the next line. By default, a carriage return command moves the pen down to the next line and over to the left margin.
-l*Left*,*Bottom*	Specifies the location on the printed page of the bottom left corner of the converted raster image. The values specified by the *Left* and *Bottom* parameters are the distances (in inches) from the bottom left corner of the printed page to the bottom left corner of the image.
-m	Enables the ``Margin 2'' mode for the 4014.
-N	Causes line feed to move the pen position down to the next line but not to the left margin. By default, a line feed command moves the pen down to the next line and over to the left margin.
-p*OutFile*	Causes the PostScript file to be written to the file specified by the *OutFile* parameter rather than the standard output.
-R	Rotates the image 90 degrees on the page for portrait orientation. The default is landscape orientation.
-s*Width*,*Height*	Specifies the size of the converted raster image on the printed page. The *Width* and *Height* parameters specify the dimensions (in inches) of the resulting image on the printed page.
-S*Width*	Allows you to scale the image without distorting its shape. The *Width* parameter specifies the width, in inches, of the resulting image on the printed page. The height of the image is computed to maintain the same ratio of height to width on the output image as on the input raster-format file.

International Character Support

See the NLSvec File for details related to character translation.

Environment Variable

PSLIBDIR Path name of a directory to use instead of the **/usr/lib/ps** for **ps4014** prologue.

Files

/usr/lib/ps/ps4014.pro Contains the default prologue file.

Related Information

NLSvec file.

---

ps630 Command

Purpose

Converts Diablo 630 print files to PostScript format.

Syntax

ps630 [**-f***Bodyfont*] [**-p***File*] [**-s***Pitch*] [**-F***Boldfont*] [*File* ...]

Description

The **ps630** command converts Diablo 630 format print files to PostScript format for printing on a PostScript printer. If no *File* variable is specified, the **ps630** command reads from standard input. By default, the PostScript file is sent to the standard output.

The **ps630** command can convert **nroff** files generated with the **-Txerox** flag. Typewheel emulation information can be specified as options. Font specifications (for bold and regular) are PostScript font names (such as Times-Roman, Times-Bold, Courier-Bold, Courier-BoldOblique). You can select 10, 12, or 15 characters per inch.

Some applications produce bold type by double-striking a character. This type of bolding is not translated into PostScript format. Only the bold effect produced by issuing the proper Diablo command sequence (Esc-O) results in bold characters.

The output of the **ps630** command cannot be page-reversed. Times-Roman and Helvetica are narrow fonts that may look squeezed if no adjustment to the page width is made by the application.

The following Diablo 630 commands are not supported:

- Print suppression
- HY-Plot
- Extended character set
- Downloading print wheel information or program mode
- Page lengths other than 11 inches
- Paper feeder control
- Hammer energy control
- Remote diagnostic
- Backward printing control.

Note: The Diablo 630 command for reverse printing is supported.

Flags

-f*Bodyfont*	Sets the font to be used for normal printing. The default is Courier.
-p*File*	Causes the PostScript file to be written to the file specified by the *File* parameter rather than to the standard output.
-s*Pitch*	Selects type size for printing (both the regular and bold fonts are scaled to this size). Pitch is in characters per inch and must be one of 10, 12, or 15. The default is 12.
-F*Boldfont*	Sets the font to be used for bold type. The default is Courier-Bold.

International Character Support

See the NLSvec File for details related to character translation.

Environment Variable

PSLIBDIR	Path name of a directory to use instead of the **/usr/lib/ps** directory for the **ps630** prologue.

Files

/usr/lib/ps	Contains PostScript fonts.

Related Information

The **enscript** command, **nroff** command.

The **NLSvec** file.

---

psc or psdit Command

Purpose

Converts **troff** intermediate format to PostScript format.

Syntax

{ **psc** | **psdit** } [**-f1** *CodeSet*:*Font*] [**-F***FontDirectory*] [**-M***MediaName*] [**-p***Prologue*] [**-o***List*] [*File*]

Description

The **psc** and **psdit** commands translate a file created by device-independent **troff** to PostScript format for printing with a PostScript printer. If no file is specified, the standard input is used. The PostScript file is sent to the standard output.

Note: The input for the **psc** and **psdit** commands should be prepared with the corresponding **-Tpsc** option, such as the **troff** or **pic** command.

The **psc** and **psdit** commands can handle extended characters created by modifying the printer code field in the font file (**/usr/lib/font/devpsc/R**). The modified field contains a string surrounded by double quotation marks. The string contains a **\b** (backslash b) followed by a sequence of characters from the standard font that is composed into a new character by overstriking.

The **psc** and **psdit** commands allow users to cause the **troff** command to include arbitrary PostScript code in the generated PostScript file. The **psc** and **psdit** commands recognize the undefined **%** (percent) command in the **troff** intermediate file format to signal the start of raw PostScript code to be placed as is in the output file. Everything between (but not including) the **%** (percent sign) and a line containing a **.** (period) will be placed in the generated PostScript output.

This PostScript output is not insulated from the **troff** command coordinate system or the state of the generated PostScript output. However, two functions are defined in the prologue so that users can insulate themselves if so desired. The **PB** (picture begin) function performs a PostScript save operation, translates the PostScript coordinate system to **troff**'s idea of the current position on the page, and changes the scale and orientation of the coordinate system axes to the standard PostScript 72 units per inch. The **PE** (picture end) macro ends this protected environment.

Several methods can be used to incorporate such included PostScript code into the **troff** intermediate file. For example, the **.sy**, **\!**, and **.cf** subcommands of the **troff** command use the following example to include the PostScript language description of a completely separate, printable document. In this example, the **showpage** operator is redefined to include `mypic.ps` as an illustration:

```
standard troff input
\&
.fl
\!%PB
\!/showpage{}def
.fl
.sy cat mypic.ps
\!PE
\!.
more standard troff input
```

Information containing various media sizes for the **psdit** command and the **enscript** command are contained in the file **/usr/lib/ps/MediaSizes**.

The information required for each entry in the **MediaSizes** file can be obtained from the **PostScript Printer Description**, or **PPD**, file that matches the PostScript printer used with TranScript. The **PPD** files are available from Adobe Systems Incorporated. The measurements extracted form the **PPD** files are in points. A printer's point is 1/72 of an inch.

Any line in the **MediaSizes** file beginning with an ASCII ***** (asterisk) is ignored when matching media size names provided on the command line to the **enscript** command and the **psdit** command.

Each entry in the **MediaSizes** file contains either eight or nine fields. The first eight fields are required for all entries. The ninth field is optional. Fields are separated by white space. The fields for each entry are as follows:

Field Name	Description
EntryName	Character string to match against a media name provided with the **-M** option with the **enscript** command or the **psdit** command.
MediaWidth	Media width in points.
MediaDepth	Media depth in points.
ImageableLLX	Imageable lower left-hand corner x coordinate in points.
ImageableLLY	Imageable lower left-hand corner y coordinate in points.
ImageableURX	Imageable upper right-hand corner x coordinate in points.
ImageableURY	Imageable upper right-hand corner y coordinate in points.
PageRegionName	PostScript sequence for the particular printer to identify the size of the imageable area.
PaperTrayName	PostScript sequence for the particular printer to select a particular paper/media tray. This field is optional.

Note: The sequence can be multiple PostScript operators or words for both the **PageRegionName** field and the **PaperTrayName** field. To specify such a sequence, use the ASCII ″ (double quotation mark character) to delimit the entire sequence.

The following are examples of field entries in the **MediaSizes** file:

Name	Width	Depth	llx	lly	urx	ury	Page- Region- Name	Paper- Tray- Name
Letter	612	792	18	17	597	776	Letter	
Legal	612	1008	18	17	597	992	Legal	

Flags

-f1 *CodeSet*:*Font*

-F*FontDirectory*	Takes font information from *FontDirectory* instead of the default.
-M*MediaName*	Specifies a media name to use to determine the amount of imageable area on the paper. The name provided is matched against entries in the **MediaSizes** file. For instance, `-M legal` would request a legal size of paper as the imageable area. If this option is not used, the default size is letter size, which is 8.5 inches wide by 11.0 inches deep.
-p*Prologue*	Uses the contents of *Prologue* instead of the default PostScript prologue.
-o*List*	Prints pages whose numbers are given in the list separated by commas. The list contains single numbers and ranges in the format *N1-N2*, where *N1* and *N2* represent page numbers. A missing *N1* means the range begins with the lowest-numbered page; a missing *N2* means the range ends with the highest-numbered page.

Examples

The following statements are equivalent:

```
pic -Tpsc File | troff -Tpsc | psc

pic -Tpsc File | troff -Tpsc | psdit
```

Environment Variables

PSLIBDIR	Path name of a directory to use instead of the **/usr/lib/ps** file for the **psc** and **psdit** command prologue.
TRANSCRIPT	Absolute path name of a file to use instead of **/usr/lib/ps/transcript.conf** for the MBCS handling.

Files

/usr/lib/font/devpsc/*	Contains the **troff** default description files for a PostScript virtual device.
/usr/lib/ps/psdit.pro	Contains the default PostScript prologue.
/usr/lib/ps/MediaSizes	Contains the default file used for media sizes.
/usr/lib/ps/transcript.conf	Contains the default value used for PostScript codeset and font name.

Related Information

The **enscript** command, **managefonts** command, **qprt** command, **pic** command, **psroff** command, **troff** command.

The **PE** macro.

me Macro Package for the nroff and troff Commands.

---

pshare Command

Purpose

Enables or reports the availability of shared login ports.

Syntax

pshare [**-a**] [*Device*]

Description

The **pshare** command enables shared ports. Shared ports are bidirectional. If you do not specify a *Device* parameter, the **pshare** command reports the names of all currently enabled shared ports. To enable a shared port, the **getty** command attempts to create a **lock** file in the **/etc/locks** directory that contains the ASCII process ID of the process. If another process is already using the port, the **getty** command waits until the port is available and tries again. The system enables a port by updating an entry in the **/etc/inittab** file and then sending a signal to the **init** process. After receiving the signal and reading the updated status entry, the process takes the appropriate action.

Use the *Device* parameter to specify the ports to be enabled. Permitted values include:

- Full device name, such as the **/dev/tty1** device
- Simple device name, such as the **tty1** device
- A number (for example, 1 to indicate the **/dev/tty1** device)

Note: You must have root user authority to run this command.

Flags

-a Enables all ports as shared.

Examples

To enable the workstation attached to the **/dev/tty2** port as a shared port, enter:

```
pshare /dev/tty2
```

/etc/inittab Controls system initialization.

Files

/etc/locks	Contains **lock** files for the **pshare** and **pdelay** commands.
/usr/sbin/pshare	Contains the **pshare** command.

Related Information

The **getty** command, **init** command, **pdelay** command, **pdisable** command, **penable** command, **phold** command, **pstart** command.

---

psplot Command

Purpose

Converts files in plot format to PostScript format.

Syntax

psplot [**-g** *Prologue*] [*File*...]

Description

The **psplot** command reads files in plot format and converts them to PostScript format on the standard output. If no files are specified, the standard input is used. The conversion is almost one-to-one, with one PostScript function call for each plot primitive. You can modify the behavior of the file by changing the definitions of the PostScript functions in the prologue.

Flags

-g*Prologue* Uses the contents of the *Prologue* file instead of the default PostScript prologue. If this flag is not specified, the default prologue file is used.

International Character Support

The **NLSvec** file provides details related to character translation.

Environment Variable

PSLIBDIR	Path name of a directory to use for the **psplot** command prologue file. The path denoted by the **PSLIBDIR** environment variable is used instead of the path denoted by the **XPSLIBDIRX** environment variable.
TRANSCRIPT	Absolute path name of a file to use instead of **/usr/lib/ps/transcript.conf** for the MBCS handling.

Files

/usr/lib/ps/psplot.pro Contains the default prologue file.

Related Information

The **ps4014** command, **lpr** command, **lp** command.

NLSvec File.

---

psrev Command

Purpose

Reverses the page order of a PostScript file and selects a page range for printing.

Syntax

psrev [**-R**] [**-s***Pagespec*,...] [*File*]

Description

The **psrev** command reverses the page order of the file specified by the *File* variable and prints the pages specified by the *Pagespec* parameter. The file must conform to PostScript file structuring conventions. If no value for the *File* is specified, the **psrev** command reads from standard input. The **psrev** command writes the resulting file to the standard output.

Flags

-R Does not reverse the page order (but subsets the pages if specified).

-s*Pagespec* Specifies a range (or several ranges) of pages to be printed. The *Pagespec* parameter is a string with no spaces. The *Pagespec* parameter can be a single page number or a range of the form *N-M*, which prints pages *N* through *M*. *-N* prints from the beginning of the document to page *N*. *M-* prints from page *M* to the end of the document.

Examples

The following are examples of using the **psrev** command showing page ranges and an individual page in nonreversed order:

```
psrev -R -s2-4,6
psrev -R -s2-4,6-8
```

Files

/var/tmp/RV* Contains the temporary file if the input is a pipe.

Related Information

The **enscript** command.

---

psroff Command

Purpose

Converts files from **troff** format to PostScript format.

Syntax

psroff [**-t**] [**-d***Queue*] [**-n***Number*] [**-t***Title*] [**-D***FontDirectory*] [**-F***FontFamily*] [**-P***Flag*] [*troffFlags*] [*File ...*]

Description

The **psroff** command is a shell script that runs the **troff** command in an environment to produce output on a PostScript printer. It uses the **psdit** command to convert **troff** intermediate output to PostScript format, and spools this output for printing. If no files are specified, the standard input is used.

To include arbitrary PostScript language commands or files in a **troff** document, see the **psdit** command.

PostScript Font Information

The PostScript Fonts for Transcript table shows the fonts available for the TranScript commands. The fonts are available by long name when using the **enscript** command, and by short name when using the **psroff** or **troff** commands. The following table shows the **psroff** commands (short names) used to declare a default set of fonts. The alphabetic characters are case-sensitive:

PostScript Fonts for Transcript

Long Name	Short Name	Font Family
AvantGarde-Book	ag	AvantGarde
AvantGarde-Demi	Ag	AvantGarde
AvantGarde-DemiOblique	AG	AvantGarde
AvantGarde-BookOblique	aG	AvantGarde
Bookman-Demi	Bo	Bookman
Bookman-DemiItalic	BO	Bookman
Bookman-Light	bo	Bookman
Bookman-LightItalic	bO	Bookman
Courier	C	Courier
Courier-Bold	CB	Courier
Courier-BoldOblique	CO	Courier
Courier-Oblique	CO	Courier
Garamond-Bold	Ga	Garamond
Garamond-BoldItalic	GA	Garamond
Garamond-Light	ga	Garamond
Garamond-LightItalic	gA	Garamond
Helvetica	H	Helvetica
Helvetica-Bold	HB	Helvetica
Helvetica-Oblique	HO	Helvetica
Helvetica-BoldOblique	HD	Helvetica
Helvetica-Narrow	hn	Helvetica
Helvetica-Narrow-Bold	Hn	Helvetica
Helvetica-Narrow-BoldOblique	HN	Helvetica
Helvetica-Narrow-Oblique	hN	Helvetica
LubalinGraph-Book	lu	Lubalin
LubalinGraph-BookOblique	lU	Lubalin
LubalinGraph-Demi	Lu	Lubalin

PostScript Fonts for Transcript

Long Name	Short Name	Font Family
LubalinGraph-DemiOblique	LU	Lubalin
NewCenturySchlbk	NC	NewCentury
NewCenturySchlbk-Bold	Nc	NewCentury
NewCenturySchlbk-Italic	nC	NewCentury
NewCenturySchlbk-Roman	nc	NewCentury
Optima	op	Optima
Optima-Bold	Op	Optima
Optima-BoldOblique	OP	Optima
Optima-Oblique	oP	Optima
Palatino-Bold	PB	Palatino
Palatino-BoldItalic	PX	Palatino
Palatino-Italic	PI	Palatino
Palatino-Roman	PA	Palatino
Souvenir-Demi	Sv	Souvenir
Souvenir-DemiItalic	SV	Souvenir
Souvenir-Light	sv	Souvenir
Souvenir-LightItalic	sV	Souvenir
Times-Bold	TB	Times
Times-BoldItalic	TD	Times
Times-Italic	TI	Times
Times-Roman	TR	Times
Symbol	S	(none)
ZapfChancery-MediumItalic	ZC	Zapf
ZapfDingbats	(none)	(none)

Flags

-D*FontDirectory* Finds font family directories in the specified font directory, rather than the standard font directory, which was configured in the installation procedure. It may be necessary to use both this flag and the **-F** flag to imitate the **-F** flag in the **troff** command.

-d*Queue* Causes the output to be queued to the queue specified by the *Queue* parameter. If the **-d** flag is not used, the **psroff** command queues output on the default queue, the first queue known to the **qdaemon**. This flag is recognized by the spooler print.

-F*FontFamily* Uses the specified font family for the R/I/B/BI fonts, rather than the Times default family. The Times, Courier, and Helvetica font families are defined at your site, and others are available as well. Ensure that the printer you use contains the font family you pick. This flag overrides the **troff** command **-F** flag. If you want to use the **troff** command **-F** flag, you should run the **troff** command directly or use the **-D** flag instead.

-n*Number* Causes the number of output copies specified by the *Number* parameter to be produced. The default is one. This flag is recognized by the spooler print.

-P*Flag* Passes the *Flag* parameter to the spooler. This flag is useful when a conflict exists between a spooler flag and a flag with the **psroff** command.

-t Sends the PostScript output to the standard output, rather than spooling it to a printer. This flag overrides the **troff** command **-t** flag. If you want the **troff** command **-t** flag, you should run the **troff** command directly.

-t*Title* Sets the job name for use on the first banner page. The default is to use the name of the first input file. This flag is recognized by the spooler print.

Parameters

troffFlags	Specifies standard flags available with the **troff** command.
File	Specifies the **troff** intermediate output file. The default is the standard input.

Files

/usr/share/lib/tmac/tmac.*	Contains the standard macro files.
/usr/lib/font/devpsc/*	Contains the **troff** description files for PostScript virtual device.
/usr/lib/ps/*.afm	Contains Adobe Font Metrics (AFM) files for use with the **enscript** command.
/usr/lib/ps/font.map	Contains the list of font names with their abbreviations.
/usr/lib/ps/ditroff.font	Contains font family files for the **troff** command.

Related Information

The **col** command, **enscript** command, **eqn** command, **lp** command, **managefonts** command, **nroff** command, **pic** command, **psdit** command, **refer** command, **tbl** command, **troff** command.

---

pstart Command

Purpose

Enables or reports the availability of login ports (normal, shared, and delayed).

Syntax

pstart [**-a**] [*Device*]

Description

The **pstart** command enables all ports (normal, shared, and delayed) listed in the **/etc/inittab** file. The system enables a port by updating an entry in the **/etc/inittab** file and then sending a signal to the **init** process. When the **init** process receives the signal and reads the updated status entry, it takes the appropriate action.

Use the *Device* parameter to specify the ports to be enabled. Permitted values include:

- A full device name, such as the **/dev/tty1** device
- A simple device name, such as the **tty1** device
- A number (for example, 1 to indicate the **/dev/tty1** device)

If you do not specify a *Device* parameter, the **pstart** command reports the names of all enabled ports and whether they are currently enabled as normal, shared, or delayed.

> **Note:** You must have root user authority to run this command.

Flag

-a Enables all ports (normal, shared, and delayed ports).

Examples

1. To display the names of all ports (normal, shared, and delayed) currently enabled and how they are enabled, enter:

   ```
   pstart
   ```

2. To enable all normal, shared, and delayed ports listed in the **/etc/inittab** file, enter:

   ```
   pstart -a
   ```

Files

/etc/locks	Contains **lock** files for the **pshare** and **pdelay** commands.
/usr/sbin/pstart	Contains the **pstart** command file.

Related Information

The **init** command, **pdelay** command, **pdisable** command, **penable** command, **phold** command, **pshare** command.

The **inittab** file.

---

pstat Command

Purpose

Interprets the contents of the various system tables and writes it to standard output.

Syntax

pstat [**-a**] [**-A**] [**-f**] [**-i**] [**-p**] [**-P**] [**-s**] [**-S**] [**-t**] [**-u***ProcSlot*] [**-T**] [**-U** *ThreadSlot*] [[*KernelFile*] *CoreFile*]

Description

The **pstat** interprets the contents of the various system tables and writes it to standard output. You must have root user or **system** group authority to run the **pstat** command.

Flags

-a	Displays entries in the process table.
-A	Displays all entries in the kernel thread table.
-f	Displays the file table.
-i	Displays the i-node table and the i-node data block addresses.
-p	Displays the process table.
-P	Displays runnable kernel thread table entries only.
-s	Displays information about the swap or paging space usage.
-S	Displays the status of the processors.
-t	Displays the tty structures.
-u *ProcSlot*	Displays the user structure of the process in the designated slot of the process table. An error message is generated if you attempt to display a swapped out process.
-T	Displays the system variables. These variables are briefly described in var.h.
-U *ThreadSlot*	Displays the user structure of the kernel thread in the designated slot of the kernel thread table. An error message is generated if you attempt to display a swapped out kernel thread.

Examples

1. To display the i-nodes of the system dump saved in the **dumpfile** core file, enter:

   ```
   pstat -i dumpfile
   ```

   Symbols are retrieved from the **/usr/lib/boot/unix** file.
2. To display the file table and the user structure for the process in process table slot 0 (zero) of the system currently running, enter:

   ```
   pstat -f -u 0
   ```
3. To display the tty structures for a system dump, whose core file is **dumpfile** and whose kernel is the **/usr/lib/boot/unix.back** file, enter:

   ```
   pstat -t /usr/lib/boot/unix.back dumpfile
   ```
4. To display all threads in the kernel thread table and the user structure of the thread in thread table slot 2, enter:

   ```
   pstat -A -U 2
   ```

Files

/usr/sbin/pstat	Contains the **pstat** command.
/dev/mem	Default system-image file.
/usr/lib/boot/unix	Default kernel-image file.
/usr/include/sys/*.h	Contains header files for table and structure information.

Related Information

The**ps** command, **stty** command.

---

ptfins command

Purpose

Generates an inslist for a PTF from a list of files.

Syntax

ptfins -f *FileNamesList* **-o** *Option* **-i** *InslistFile*

Description

The output file is written in the current directory with the naming convention *option***.il**. The **ptfins** command is invoked by the **ptfpkg** command. The **ptfins** command generates an inslist file from the **FileNamesList** and any entries in the *option***.inv_u** file. The **ptfins** command searches the inslist file specified with the **-i** flag for the inslist entry associated with each file listed in **FileNamesList**. The inslist entry is included in the output inslist file.

The *option***.inv_u** file contains inslist entries for inventory-only changes. Inventory-only changes are modifications that do not require a file to be shipped on the media. An example of an inventory-only change is a modification to the attribute or link information for a shipped file. If the file itself is not modified, then a new entry must be created in the inventory file for the update but it is not necessary to reship the file. The **ptfins** command reads the entries in the *option***.inv_u** and writes an inventory-only entry to the output inslist file.

Flags

-f *FileNamesList*	Specifies the location of a file containing the names of the shipped files to be included in the update.
-i *InslistFile*	Specifies the location of the complete inslist file for the option.
-o *Option*	Specifies the name of the option or fileset.

Note: All flags are required.

Environment Variables

BLDCYCLE	Specifies the current build cycle. The build-cycle identifier identifies the build in which the PTF was created.
TOP	Specifies that the **$TOP** environment variable is set to the base of the update tree. The **PTF**, **HISTORY**, and **UPDATE** directories exist relative to **$TOP**.

Example

To generate an inslist file for option **bos.adt.cls.lib**, type:

```
ptfins -f ./FileNamesList -o bos.adt.cls.lib -i $TOP/UPDATE/$OPTION/bos.adt.cls.lib.il
```

Input Files

FileNamesList	Specifies a list of shipped file names, one per line. The files listed in **FileNamesList** are those shipped files included in the update. The name of the file matches the name in the inslist file specified with the **-i** flag.
InslistFile	Specifies the inslist file for the option or fileset. The **ptfins** command searches the inslist for each file listed in the **FileNamesList**.
$TOP/UPDATE/$OPTION/*option***.inv_u**	Specifies inslist entries that have been modified for an inventory-only update. The **ptfins** command reads the *option***.inv_u** file and writes the modified inslist entry to its output file as an inventory-only entry. This file is optional.

Output File

*option***.il**	Specifies that the output file is an inslist with entries for all files in **FileNamesList** and all entries in *option***.inv_u**.

Related Information

The **ptfpkg** command.

---

ptfpkg Command

Purpose

Creates a PTF image for each PTF listed in the input **ptf_pkg.$BLDCYCLE** file.

Syntax

ptfpkg -c -b [**-m** *PtfType*] [**-d** *OutputDirectory*]

Description

The output image can be installed using the **installp** command.

The **ptfpkg** command calculates a new version.release.mod.fix (VRMF) level for each PTF. Each fileset PTF increments the fix level value by 1. Each **cum_ptf** increments the mod level value by one and resets the fix level to 0.

To override the default VRMF, the **fileset** field of the **PTF_PKG** file can be modified to include a new VRMF value. This is typically only used for generating a **cum_ptf** with a non-zero fix level. The **fileset** field of the **PTF_PKG** would contain the following format:

FilesetName:v.r.m.f

where the **fileset** field is the fourth field of each line in the **PTF_PKG** file.

The **ptfpkg** command processes PTFs by LPP. The **ptfpkg** command reads the **LPPLIST** file for each build cycle. Each LPP in this file has a **PTF_PKG** file. The **ptfpkg** command reads the **PTF_PKG** file for each LPP and generates a AIX Version 4 update image for each PTF in the **PTF_PKG** file.

All regular fileset PTFs are packaged in the **$TOP/UPDATE/$OPTION/$PTF** directory. The **ptfpkg** command creates the **liblpp.a** files appropriate for the update, generates the **lpp_name** file, and generates a backup format file (bff) or ccss image containing all files listed in the **PTF_PKG** file.

The packaging directory for special PTFs is **$TOP/UPDATE/$LPP/$PTF**. Special PTFs are indicated by a file name of **pkg_ptf**, **pmp_ptf**, **enh_ptf**, **C_ptf**, or **opp_ptf**. These types of updates are enabled at a later date.

After a PTF is created, the **ptfpkg** command can repackage the PTF only if one of the following conditions is satisfied:

- A special PTF can always be repackaged.
- A cumulative PTF can be repackaged if the PTF is the last entry in the **CUMLIST** file or if the PTF is not in the **CUMSLIST** file and the file **PTFSLIST** is empty.
- All other PTFs can be repackaged if the PTF is not in the **PTFSLIST** file or if the **PTF** is in the **PTFSLIST** file and all entries after the PTF are in the **PTF_PKG** file.

Each time a PTF is successfully built, it is added to the **PTFSLIST** file if it is a fileset PTF or to the **CUMSLIST** file if it is a cumulative PTF. If the **$TOP/PTF/$BLDCYCLE/defectapars** information exists, the PTF is also added to the **ptfapardef.master** file.

Pseudocode

```
for each LPP listed in lpplist.$BLDCYCLE
          for each PTF listed in the ptf_pkg.$BLDCYCLE
               if PTFDIR exists then repackage
               call getvrmf command to calculate new VRMF value and generate internal VRMF table
               call updatefixdata command to update the fixdata database
               call processPtf command to generate PTF list files and option.prereq file
               call lookupfixdata command to generate option.fixdata file
               call ptfins command to generate inslist file for update
               call adeinv command to generate .inventory, .al, .tcb and .size files
               call adelppname command to generate lpp_name file
               call adepackage command to create bff image
               if -c option used to generate ccss image
                    call gen_toc_entry to generate toc file
                    call gen_infofile to generate infofile
                    call ccss_pack to generate ccss image
               fi
```

```
            update $TOP/HISTORY/ptfoptions file to include PTF, fileset and vrmf information
            create ptfapardef.master file from $TOP/PTF/*/defectapars files if they exist
        done
    done
```

Flags

-b Specifies that the bff images are not deleted.
Note: One or both of the **-b** and **-c** options is required.

-c Creates PTF images in ccss format.
Note: One or both of the **-b** and **-c** options is required.

-d *OutputDirectory* Specifies a directory where the bff or ccss images are copied.

-m *PtfType* Specifies the PTF type. Use this flag to override the default type value calculated by the **ptfpkg** command. If the **-m** flag is not specified, the **ptfpkg** command calculates the PTF type based on the **file name** field of the **ptf_pkg** file as indicated:

- If the file name is **pkg_ptf**, the package type is **M** (maintenance level packaging PTF).
- If the file name is **pmp_ptf**, the package type is **P** (packaging PTF).
- If the file name is **opp_ptf** or **C_ptf**, the package type is **MC** (cumulative maintenance packaging PTF).
- If the file name is **enh_ptf**, the package type is **ME** (enhancement packaging PTF).
- If the file name is **/usr/sbin/installp**, the package type is **SF** (fileset PTF required to install first).
- If none of the above conditions is met and the user has not overridden the PTF type with the **-n** option, the PTF type defaults to **S** for single PTF or fileset update.

Environment Variables

The following environment variables are required for running the **ptfpkg** command:

BLDCYCLE Specifies the current build cycle. The build-cycle identifier identifies the build in which the PTF was created.

BUILD_TYPE Specifies the type of environment this build is occurring in. Valid values are **production**, **area**, and **sandbox**, where sandbox builds are development level builds. The **ptfpkg** command requires valid APAR text in the **$TOP/HISTORY/abstracts** file for each APAR if the build type is production.

ODE_TOOLS Specifies the path to the build tools. The tools are located in the **usr/bin** directory relative to the location specified by **$ODE_TOOLS**. The **PATH** environment variable includes **$ODE_TOOLS/usr/bin**.

SHIP_PATH Specifies the colon-separated list of paths to ship trees. Ship trees are searched in the order specified.

TOP Specifies the full pathname of the top of the tree from which the build is initiated. The HISTORY, UPDATE, and PTF trees exist relative to **$TOP**.

Warnings

Warning messages are informational and do not affect the return code of **ptfpkg**.

Errors

Error messages are produced when **ptfpkg** encounters an error during processing. Work is stopped on the current LPP or PTF, and processing of the next LPP or PTF is started. Return codes are not affected. The PTFs that had an error are left in the **ptf_pkg** file so the PTF can be packaged after the errors are corrected.

Fatal Errors

Fatal Error messages are produced when **ptfpkg** encounters an error that cannot be recovered from. Processing terminates with a non-zero return code.

Examples

1. To call **ptfpkg** with no arguments, media type defaulting to single update **S**, and shipping the resulting bff images into the **$TOP/update.bff.images** directory, type:

   ```
   ptfpkg -d $TOP/update.bff.images
   ```

2. To call **ptfpkg** setting the media type to packaging update **P**, generating ccss images instead of bff images, and placing them in the **$TOP/ptf.images** directory, type:

   ```
   ptfpkg -m P -c -d $TOP/ptf.images
   ```

Files

The following files must be provided by the user:

data.liblpp.a.$BLDCYCLE
Specifies that if this file exists in the option directory, the **ptfpkg** command assumes the fileset is a share data fileset. If the **data.liblpp.a.$BLDCYCLE** file does not exist in the option directory, then **ptfpkg** looks for the **data.liblpp.a** file. It makes a copy of the file adding the build-cycle suffix. This file is an archive containing any odm scripts or config scripts for the update. The **mkodmupdt** command can be used to generate the odm scripts if applicable.

lpp_info
Contains the FORMAT, PLATFORM, and PRODNAME. FORMAT is the update format type, usually 4 for AIX Version 4 updates. PLATFORM is R for RS/6000. PRODNAME is the product name of the package of which the fileset is a member. If the **lpp_info.$BLDCYCLE** file does not exist in the option directory, then **ptfpkg** looks for the **usr.liblpp.a** file. It makes a copy of the file adding the build-cycle suffix.

lpplist.$BLDCYCLE
Contains the list of LPPs that has **ptf_pkg** files for the current build cycle. The format of this file is one LPP name per line.

*option***.il.$BLDCYCLE**
Specifies the file used by the **ptfins** command to generate the inslist information for each file that is shipped with the update. If the *option***.il.$BLDCYCLE** file does not exist in the option directory, then **ptfpkg** looks for the *option***.il** file. It makes a copy of the file adding the build-cycle suffix.

*option***.lp.$BLDCYCLE**
Specifies the file containing the default BOOT, CONTENT, and LANG information for the option. If the *option***.lp.$BLDCYCLE** file does not exist in the option directory, then **ptfpkg** looks for the *option***.lp** file. It makes a copy of the file adding the build-cycle suffix.

ptf_pkg.$BLDCYCLE
Contains the packaging information for all PTFs and filesets associated with the LPP. If a single fileset PTF ships multiple files, then the **ptf_pkg** file contains one line for each fileset. The format of the **ptf_pkg** file is:

ptf | *apars list* | *filename* | *fileset* | *ifreqs* | *coreqs* | *prereqs*

root.liblpp.a.$BLDCYCLE
Specifies that if this file exists in the option directory, the **ptfpkg** command assumes the fileset contains a root part. This file exists for options with a root part even if it is an archive file with no members. If the root **liblpp.a** file for the update contains no members after processing is completed, then **ptfpkg** sets the CONTENT value back to U (`usr` only). If the **root.liblpp.a.$BLDCYCLE** file does not exist in the option directory, then **ptfpkg** looks for the **root.liblpp.a** file. It makes a copy of the file adding the build-cycle suffix. This file is an archive containing any odm scripts or config scripts for the update. The **mkodmupdt** command can be used to generate the odm scripts if applicable.

usr.liblpp.a.$BLDCYCLE	Specifies that if this file exists in the option directory, the **ptfpkg** command assumes the fileset contains a `usr` part. If the **data.liblpp.a** file does not exist for the option, then the **usr.liblpp.a** file is required to exist. If the **usr.liblpp.a.$BLDCYCLE** file does not exist in the option directory, then **ptfpkg** looks for the **usr.liblpp.a** file. It makes a copy of the file adding the build-cycle suffix. This file is an archive containing any odm scripts or config scripts for the update. The **mkodmupdt** command can be used to generate the odm scripts if applicable.

The **ptfpkg** command uses the following files and directories to generate AIX Version 4 PTFs:

***CUMSLIST**	**$TOP/UPDATE/$OPTION/cumsList**
INSLIST	**$TOP/UPDATE/$OPTION/***option***.il.$BLDCYCLE**
LPFILE	**$TOP/UPDATE/$OPTION/***option***.lp.$BLDCYCLE**
LPPINFO	**$TOP/UPDATE/$OPTION/lpp_info.$BLDCYCLE**
LPPLIST	**$TOP/UPDATE/lpplist.$BLDCYCLE**
PTF_PKG	**$TOP/UPDATE/$LPP/ptf_pkg.$BLDCYCLE**
PTFDIR	**$TOP/UPDATE/$OPTION/$PTFNUMBER**
***PTFSLIST**	**$TOP/UPDATE/$OPTION/ptfsList**
USRLIBLPPA	**$TOP/UPDATE/$OPTION/usr.liblpp.a.$BLDCYCLE**
ROOTLIBLPPA	**$TOP/UPDATE/$OPTION/root.liblpp.a.$BLDCYCLE**
DATALIBLPPA	**$TOP/UPDATE/$OPTION/data.liblpp.a.$BLDCYCLE**

Note: The files that are marked with an * (asterisk) are generated and maintained by the **ptfpkg** command and are not provided by the user.

Related Information

The **adeinv** command, **adelppname** command, **adepackage** command, **getvrmf** command, **installp** command, **lookupfixdata** command, **mkodmupdt** command, **processPtf** command, **ptfins** command, **updatefixdata** command.

---

ptx Command

Purpose

Generates a permuted index.

Syntax

ptx [**-f**] [**-r**] [**-t**] [**-b** *Breakfile*] [**-g** *Number*] [**-w** *Number*] [**-i** *Ignore* | **-o** *Only*] [**—**] [*Infile* [*Outfile*]]

Description

The **ptx** command reads the specified English-language text (the *Infile* parameter), creates a rearranged index from it, and writes to the specified file (*Outfile*). Standard input and standard output are the defaults.

The **ptx** command searches the specified file (*Infile*) for keywords, sorts the lines, and generates the file *Outfile*. The *Outfile* file can then be processed with the **nroff** or **troff** command to produce a rearranged index.

The **ptx** command follows three steps:

1. Performs the permutation, generates one line for each keyword in an input line, and rotates the keyword to the front of the line.

2. Sorts the permuted file.
3. Rotates the sorted lines so that the keyword comes at the middle of each line.

The resulting lines in the *Outfile* file are in the following form:

```
.xx "" "before keyword" "keyword" "after keyword"
```

where `.xx` is an **nroff** or **troff** macro provided by the user or by the **ptx** command. The **mptx** macro package provides the `.xx` macro definition.

The `before keyword`, and `keyword`, and `after keyword` fields incorporate as much of the line as can fit around the keyword when it is printed. The first field and last field, at least one of which is always the empty string, are wrapped to fit in the unused space at the opposite end of the line.

Notes:

1. Line-length counts do not account for overstriking or proportional spacing.
2. Lines that contain a ~ (tilde) do not work, because the **ptx** command uses that character internally.
3. The **ptx** command does not discard non-alphanumeric characters.

Flags

-b *BreakFile*	Uses the characters in the specified break file to separate words. Tab characters, new-line characters, and spaces are always used as break characters.
-f	Folds uppercase and lowercase characters for sorting.
-g *Number*	Uses the specified number as the number of characters that the **ptx** command reserves for each gap among the four parts of the line as it is printed. The default *Number* variable value is 3.
-i *Ignore*	Does not use any words specified in the *Ignore* file as keywords. If the **-i** and **-o** flags are not used, the **/usr/lib/eign** file is the default *Ignore* file.
-o *Only*	Uses only the words specified in the *Only* file as keywords.
-r	Considers any leading non-blank characters of each input line as reference identifiers separate from the text of the line. Attaches the identifier as a fifth field on each output line.
-t	Prepares the output for the phototypesetter.
-w *Number*	Uses the specified number as the length of the output line. The default line length is 72 characters for the **nroff** command and 100 for the **troff** command.
—	(double dash) Indicates the end of flags.

Parameters

Infile	Specifies the English-language text. Standard input is the default file. The **ptx** command searches the specified file for keywords, sorts the lines, and generates the file *Outfile*.
Outfile	Specifies the file to which the **ptx** command writes the index created from the *Infile* file. Standard output is the default file. The *Outfile* file can be processed with the **nroff** or **troff** command to produce a rearranged index.

Files

/usr/lib/eign	Contains the default *Ignore* file.
/usr/share/lib/tmac/tmac.ptx	Contains the macro file.

Related Information

The **nroff** command, **troff** command.

The **mm** macro package, **mptx** macro package**.**

---

pwchange Command

Purpose

Change user authentication and privacy keys dynamically.

Syntax

pwchange [**-e**] [**-d** *DebugLevel*] [**-p** *Protocol*] [**-u** *KeyUsage*] [**-s**] [*OldPassword NewPassword*] [*IPAddress* | *HostName* | *EngineID*]

Description

The **pwchange** command is provided to facilitate dynamic changes of user authentication and privacy keys. Dynamic configuration of authentication and privacy keys is done by doing **set** commands to objects of syntax keyChange. The keyChange syntax provides a way of changing keys without requiring that the actual keys (either new or old) be flowed directly across the wire, which would not be secure. Instead, if an object,such as **usmUserAuthKeyChange** (for example) is to be set, the keyChange value must be derived from the old and new passwords and the engineID of the agent at which the key will be used. The **pwchange** command is used to generate the keyChange values.

The **pwchange** command generates different output, depending on which protocol and what key usage is selected. Keychange values are typically twice as long as the key to be changed.

Flags

-d *DebugLevel*
: This flag indicates what level of debug information is desired. Debug tracing is either on or off: 1 causes debug tracing to be generated to the screen of the command issuer (sysout). Debug tracing is off (0) by default.

-e
: This flag indicates that the agent for which the keychange value is being defined is identified by engineID rather than by IP address or host name.

-p *Protocol*
: This flag indicates the protocols for which the keychange values should be generated. Valid values are:

  HMAC-MD5
  : Generates keychange values for use with the HMAC-MD5 authentication protocol.

  HMAC-SHA
  : Generates keychange values for use with the HMAC-SHA authentication protocol.

  all
  : Generates both HMAC-MD5 and HMAC-SHA keychange values.

  The default is that keychange values for the HMAC-MD5 protocol are generated.

-s
: This flag indicates that output should be displayed with additional spaces to improve readability. By default, data is displayed in a condensed format to facilitate cut-and-paste operations on the keychange values onto command lines in shell scripts.

-u *KeyUsage*

This flag indicates the usage intended for the keychange value. Valid values are:

auth An authentication keychange value.

priv A privacy keychange value.

all Both authentication and privacy keychange values.
Note: There is no difference between a keychange value generated for authentication and a keychange value generated for privacy. However, the length of privacy keychange values depends on whether the keychange value is localized.

Parameters

EngineID Specifies the engineID (1-32 octets, 2-64 hex digits) of the destination host at which the key is to be used. The engineID must be a string of 1-32 octets (2-64 hex digits). The default is that the agent identification is not an engineID.

HostName Specifies the destination host at which the key is to be used.

IPAddress Specifies the IP address in dotted decimal notation of the agent at the destination host at which the key is to be used.

NewPassword Specifies the password that will be used in generating the new key. The password must be between eight and 255 characters long.

OldPassword Specifies the password that was used in generating the key originally. The password must be between eight and 255 characters long.

Examples

The **pwchange** command generates different output depending on which protocol and what key usage is selected. Key change values are typically twice as long as the key to be changed.

1. The following command demonstrates how pwchange can be used:

```
pwchange oldpassword newpassword 9.67.113.79
```

   The output of this command looks similar to:

```
Dump of 32 byte HMAC-MD5 authKey keyChange value:
  3eca6ff34b59010d262845210a401656
  78dd9646e31e9f890480a233dbe1114d
```

   The value to be set should be passed as a hex value with the **clsnmp** command (all on one line):

```
clsnmp set usmUserAuthKeyChange.12.0.0.0.2.0.0.0.0.9.67.113.79.2.117.49
\'3eca6ff34b59010d262845210a40165678dd9646e31e9f890480a233dbe1114d\'h
```

   Note: The backslash in the preceding example is required before the single quotation mark to enable AIX to correctly interpret the hexadecimal value.

   The index of the usmUserTable is made up of the EngineID and the ASCII representation of the user name. In this case it is 2 characters long and translates to 117.49.

   Note: **pwchange** incorporates a random component in generating keys and keyChange values. The output from multiple commands with the same input does not produce duplicate results.

Related Information

The **clsnmp** command, **pwtokey** command, **snmpdv3** daemon.

The **/etc/clsnmp.conf** file, **/etc/snmpdv3.conf** file.

---

pwck Command

Purpose

Verifies the correctness of local authentication information.

Syntax

pwck

Description

The **pwck** command verifies the correctness of the password information in the user database files by checking the definitions for all users. The **pwck** command internally calls the **pwdck** command with **-n** and **ALL** options.

Exit Status

0 The command completed successfully.

>0 An error occurred.

Examples

1. To verify that all the users and administrators exist in the user database, and have any errors reported (but not fixed), enter:

   ```
   pwck
   ```

Files

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

Related Information

The **pwdck** command.

---

pwd Command

Purpose

Displays the path name of the working directory.

Syntax

pwd

Description

The **pwd** command writes to standard output the full path name of your current directory (from the root directory). All directories are separated by a / (slash). The root directory is represented by the first /, and the last directory named is your current directory.

Exit Status

This command returns the following exit values:

0 Successful completion.
>0 An error occurred.

Examples

Entering:

```
pwd
```

displays the current directory as:

```
/home/thomas
```

Files

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

Related Information

The **cd** command.

The **getcwd** subroutine.

Directory Overview in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices*.

---

pwdadm Command

Purpose

Administers users' passwords.

Syntax

pwdadm [**-f** *Flags* | **-q** | **-c**] *User*

Description

The **pwdadm** command administers users' passwords. The root user or a member of the security group can supply or change the password of the user specified by the *User* parameter. The invoker of the command must provide a password when queried before being allowed to change the other user's password. When the command executes, it sets the **ADMCHG** attribute. This forces the user to change the password the next time a **login** command or an **su** command is given for the user.

Root users and members of the security group should not change their personal password with this command. The **ADMCHG** attribute would require them to change their password again the next time a **login** command or an **su** command is given for the user. Only the root user or a user with PasswdAdmin authorization can change password information for administrative users, who have the **admin** attribute set to true in the **/etc/security/user** file.

Only the root user, a member of the security group, or a user with PasswdManage authorization can supply or change the password of the user specified by the *User* parameter.

When this command is executed, the password field for the user in the **/etc/passwd** file is set to ! (exclamation point), indicating that an encrypted version of the password is in the **/etc/security/passwd** file. The **ADMCHG** attribute is set when the root user or a member of the security group changes a user's password with the **pwdadm** command.

A new password must be defined according to the rules in the **/etc/security/user** file, unless the **-f NOCHECK** flag is included. Only 7-bit characters are supported in passwords. By including the **-f** flag with the **pwdadm** command, the root user or a member of the security group can set attributes that change the password rules. If there is no password entry in the **/etc/security/passwd** file when the **-f** flag is used, the password field in the **/etc/passwd** file is set to ! (exclamation point) and an * (asterisk) appears in the password= field to indicate that no password has been set.

The **-q** flag permits the root user or members of the security group to query password information. Only the status of the **lastupdate** attribute and the **flags** attribute appear. The encrypted password remains hidden.

The **-c** flag clears all flags set in the **/etc/security/passwd** file.

Flags

-f *Flags* Specifies the **flags** attribute of a password. The *Flags* variable must be from the following list of comma-separated attributes:

NOCHECK
Signifies that new passwords need not follow the guidelines established in the **/etc/security/user** file for password composition.

ADMIN Specifies that password information may be changed only by the root user. Only the root user can enable or disable this attribute.

ADMCHG
Resets the **ADMCHG** attribute without changing the user's password. This forces the user to change passwords the next time a **login** command or an **su** command is given for the user. The attribute is cleared when the user specified by the *User* parameter resets the password.

-q Queries the status of the password. The values of the **lastupdate** attribute and the **flags** attribute appear.

-c Clears all flags set in the **/etc/security/passwd** file.

Security

Access Control: Only the root user and members of the security group should have execute (x) access to this command. The command should have the **trusted computing base** attribute and be **setuid** to the root user to have write (w) access to the **/etc/passwd** file, the **/etc/security/passwd** file, and other user database files.

Files Accessed:

Mode	File
rw	**/etc/passwd**
rw	**/etc/security/passwd**
r	**/etc/security/user**

Auditing Events:

Event	Information
PASSWORD_Change	user
PASSWORD_Flags	user, flags

Examples

1. To set a password for user `susan`, a member of the security group enters:

   ```
   pwdadm susan
   ```

   When prompted, the user who invoked the command is prompted for a password before Susan's password can be changed.
2. To query the password status for user `susan`, a member of the security group enters:

   ```
   pwdadm -q susan
   ```

   This command displays values for the **lastupdate** attribute and the **flags** attribute. The following example shows what appears when the **NOCHECK** and **ADMCHG flags** attributes are in effect:

   ```
   susan:
           lastupdate=
           flags= NOCHECK,ADMCHG
   ```

Files

/usr/bin/pwdadm	Contains the **pwdadm** command.
/etc/passwd	Contains the basic user attributes.
/etc/security/passwd	Contains password information.
/etc/security/login.cfg	Contains configuration information.

Related Information

The **passwd** command.

For more information about the identification and authentication of users, discretionary access control, the trusted computing base, and auditing, refer to Standalone System Security in *AIX 5L Version 5.2 Security Guide*.

---

pwdck Command

Purpose

Verifies the correctness of local authentication information.

Syntax

pwdck { **-p** | **-n** | **-t** | **-y** } { **ALL** | *User ...* }

Description

The **pwdck** command verifies the correctness of the password information in the user database files by checking the definitions for **ALL** the users or for the users specified by the *User* parameter. If more than one user is specified, there must be a space between the names.

> **Note:** This command writes its messages to **stderr**.

You must select a flag to indicate whether the system should try to fix erroneous attributes. The following attributes are checked for locally defined users in the **/etc/passwd** file:

entry Ensures that each entry is readable and that it contains at least two : (colons). If you indicate that the system should fix errors, the entire entry is discarded.

passwd Ensures that the password field is an ! (exclamation point). If you indicate that the system should fix errors, it transfers the information in the password field to the **/etc/security/passwd** file, updates the **lastupdate** attribute in the **/etc/security/passwd** file, and then replaces the password field in the **/etc/passwd** file with an !. In general, passwords are required if the **minalpha**, **minother**, or **minlen** password restriction is set to a nonzero value in the **/etc/security/user** file.

user Ensures that the user name is a unique string of 8 bytes or less. It cannot begin with a + (plus sign), a : (colon), a - (minus sign), or a ~ (tilde). It cannot contain a : (colon) in the string and cannot be the **ALL**, **default**, or ***** keywords. If you indicate that the system should fix errors, it removes this user's entry line from the **/etc/passwd** file. If the user name starts with a + or a - symbol, the user is not locally defined, and checks are not performed.

Attributes checked in the **/etc/security/passwd** file are:

line Ensures that each line is readable and is part of a stanza. Any invalid line is discarded.

password Ensures that the **password** attribute exists and is not blank, if passwords are required on the system. If you indicate that the system should fix errors, the password is set to ***** (asterisk), and the **lastupdate** attribute is discarded.

In general, passwords are required if either of the **minalpha** or **minother** password restrictions are set to nonzero values in the **/etc/security/user** file. If a user's **flags** attribute specifies the **NOCHECK** keyword, a password is not required for this user, and the check is ignored.

lastupdate Ensures that the **lastupdate** attribute exists for a valid non-blank password, and that its time is prior to the current time. If you indicate that the system should fix errors, the **lastupdate** attribute is discarded or updated, depending on the **password** attribute. The **lastupdate** attribute is discarded if the **password** attribute doesn't exist, or equals a blank or an * (asterisk). Otherwise, the **lastupdate** time is set to the current time.

flags Ensures that the **flags** attribute contains only the keywords **ADMIN**, **ADMCHG**, and **NOCHECK**. If you indicate that the system should fix errors, it deletes any undefined flags.

Attributes checked in the **/etc/security/user** file are:

auth1 Ensures that each SYSTEM;*username* entry defined for a local user has an *username* entry in the **/etc/security/passwd** file. If you indicate that the system should fix errors, a stanza is added to the **/etc/security/passwd** file for each missing entry, in the following format:

```
username:
          password = *
```

If a user's entry and a default entry both are missing from the **/etc/security/user** file, the system assumes the following values and the check on auth1 is performed:

```
auth1 = SYSTEM;user
```

auth2 Ensures that each authname;*username* entry defined for a local user has an *username* entry in the **/etc/security/passwd** file. If you indicate that the system should fix errors, an entry is added for each missing entry.

If a user's entry and a default entry both are missing from the **/etc/security/user** file, the system assumes the following values and the check on **auth2** is performed:

```
auth2 = NONE
```

When ALL is specified, the **pwdck** command ensures that each stanza in the **/etc/security/passwd** file corresponds to an authentication name of a local user as a SYSTEM;*username* entry in the **/etc/security/user** file. If you indicate that the system should fix errors, a stanza which does not correspond to an username entry in the **/etc/security/user** file is discarded from the **/etc/security/passwd** file.

The **pwdck** command locks the **/etc/passwd** file and the **/etc/security/passwd** file when it updates them. If either of these files are locked by another process, the **pwdck** command waits a few minutes for the files to be unlocked, and terminates if this does not happen.

The **pwdck** command checks to see if the **/etc/passwd** file and the **/etc/security/passwd** file are modified by another process while the current **pwdck** process is running. If you indicate that the system should fix errors, the **pwdck** command updates the **/etc/passwd** file and the **/etc/security/passwd** file, and may overwrite any changes made by the other process.

The **pwdck** command also checks to see if the database management security files (**/etc/passwd.nm.idx**, **/etc/passwd.id.idx**, **/etc/security/passwd.idx**, and **/etc/security/lastlog.idx**) files are up-to-date or newer than the corresponding system security files. Please note, it is alright for the **/etc/security/lastlog.idx** to be not newer than **/etc/security/lastlog**. If the database management security files are out-of-date, a warning message appears indicating that the root user should run the **mkpasswd** command.

Generally, the **sysck** command calls the **pwdck** command as part of the verification of a trusted-system installation. In addition, the root user or a member of the security group can enter the command.

Flags

-n Reports errors but does not fix them.
-p Fixes errors but does not report them.
-t Reports errors and asks if they should be fixed.
-y Fixes errors and reports them.

Security

Access Control: This command should grant execute (x) access to the root user and members of the security group. The command should be **setuid** to the root user, to read and write the authentication information, and have the **trusted computing base** attribute.

Files Accessed:

Mode	File
rw	/etc/passwd
r	/etc/security/user
rw	/etc/security/passwd
r	/etc/security/login.cfg

Auditing Events:

Event	Information
PASSWORD_Check	user, error/fix, status

Event	Information
PASSWORD_Ckerr	file/user, error, status

Examples

1. To verify that all local users have valid passwords, enter:

   ```
   pwdck  -y ALL
   ```

   This reports errors, and fixes them.
2. To ensure that user `ariel` has a valid stanza in the **/etc/security/passwd** file, enter:

   ```
   pwdck  -y ariel
   ```

Files

/usr/bin/pwdck	Contains the **pwdck** command.
/etc/passwd	Contains the basic user attributes.
/etc/security/passwd	Contains actual passwords and security information.
/etc/security/user	Contains the extended attributes of users.
/etc/security/login.cfg	Contains configuration information and password restrictions.

Related Information

The **grpck** command, **mkpasswd** command, **sysck** command, **usrck** command.

Standalone System Security in *AIX 5L Version 5.2 Security Guide* describes the identification and authentication of users, discretionary access control, the trusted computing base, and auditing.

---

pwtokey Command

Purpose

Converts passwords into localized and nonlocalized authentication and privacy keys.

Syntax

pwtokey [**-e**] [**-d** *DebugLevel*] [**-p** *Protocol*] [**-u** *KeyUsage*] [**-s**] *Password* [*EngineID* | *HostName* | *IPAddress*]

Description

AIX provides a facility called **pwtokey** that allows conversion of passwords into localized and nonlocalized authentication and privacy keys. The **pwtokey** procedure takes as input a password and an identifier of the agent and generates authentication and privacy keys. Since the procedure used by the pwtokey facility is the same algorithm used by the **clsnmp** command, the person configuring the SNMP agent can generate appropriate authentication and privacy keys to put in the **snmpd.conf** file for a user, given a particular password and the IP address at which the agent will run.

If the IP address or the hostname is specified, the SNMP agent must be an AIX agent. The engineID will be created using a vendor-specific formula that incorporates the IP address of the agent and an enterprise ID representing AIX.

Flags

-d *DebugLevel*
This flag indicates what level of debug information is desired. Debug tracing is either on or off, so a value of 1 causes debug tracing to be generated to the screen of the command issuer (sysout), and a value of 0 specifies that no debug tracing be generated. Debug tracing is off (0) by default.

-e
This flag indicates that the agent for which the key is being defined is identified by engineID rather than by IP address or host name.

-p *Protocol*
This flag indicates the protocols for which the keys should be generated. Valid values are:

HMAC-MD5
Generates keys for use with the HMAC-MD5 authentication protocol.

HMAC-SHA
Generates keys for use with the HMAC-SHA authentication protocol

all Generates both HMAC-MD5 and HMAC-SHA keys.

The default is that keys for the HMAC-MD5 protocol are generated.

-s
This flag indicates that output data should be displayed with additional spaces to improve readability. By default, data is displayed in a condensed format to facilitate cut-and-paste operations on the keys into configuration files or command lines.

-u *KeyUsage*
This flag indicates the usage intended for the key. Valid values are:

auth An authentication key.

priv A privacy key.

all Both authentication and privacy keys.
Note: There is no difference between a key generated for authentication and a key generated for privacy. However, the length of privacy keys depends on whether the key is localized or not.

Parameters

EngineID
Specifies the engineID of the SNMP agent at which the key will be used. The engineID is determined at SNMP agent initialization from the snmpd.boots file. The engineID must be a string of 1-32 octets (2-64 hex digits). The default is that the agent identification is not an engineID.

HostName
Specifies the SNMP agent at which the key will be used on an SNMP request.

IPAddress
Specifies the IP address in dotted decimal notation of the SNMP agent at which the key will be used on an SNMP request.

Password

Specifies the text string to be used in generating the keys. The password must be in the range of 8-255 characters long. In general, while any printable characters can be used in the passwords, the AIX shell may interpret some characters rather than passing them to the pwtokey command. Include passwords in single quotes to avoid interpretation of the characters by the AIX shell.
Note: This password is not related to the community name (or ″password″) used with community-based security (SNMPv1 and SNMPv2c). This password is used only to generate keys for user-based security, an entirely different security scheme.

Examples

1. This example shows a simple invocation of the pwtokey command:

   ```
   pwtokey testpassword 9.67.113.79
   ```

   The output from this command looks similar to the following:

   ```
   Display of 16 byte HMAC-MD5 authKey:
    775b109f79a6b71f94cca5d22451cc0e

   Display of 16 byte HMAC-MD5 localized authKey:
    de25243d5c2765f0ce273e4bcf941701
   ```

   As this example shows, **pwtokey** generates two keys—one that is localized (has been tailored to be usable only at the agent identified) and one that has not been localized. Typically, the localized key is used in the configuration for the SNMP agent. The nonlocalized key is used in the configuration for the clsnmp command.
2. The **pwtokey** can be invoked requesting HMAC-SHA keys for both authentication and privacy, as in the following example:

   ```
   pwtokey -p HMAC-SHA -u all testpassword 9.67.113.79
   ```

   The output of this command looks similar to the following:

   ```
   Display of 20 byte HMAC-SHA authKey:
    b267809aee4b8ef450a7872d6e348713f04b9c50

   Display of 20 byte HMAC-SHA localized authKey:
    e5438092d1098a43e27e507e50d32c0edaa39b7c

   Display of 20 byte HMAC-SHA privKey:
    b267809aee4b8ef450a7872d6e348713f04b9c50

   Display of 16 byte HMAC-SHA localized privKey:
    e5438092d1098a43e27e507e50d32c0e
   ```

   The output for the privacy keys is the same as the output for the authentication keys, except that the localized privacy key has been truncated to 16 bytes, as is required for DES.

   Note: If encryption is used, it is more secure to use different passwords for authentication and privacy.

Related Information

The **clsnmp** command, **pwchange** command, **snmpdv3** daemon.

The **/etc/clsnmp.conf** file, **/etc/snmpdv3.conf** file.

pxed Command

Purpose

Implements a Preboot Execution Environment (PXE) Proxy Dynamic Host Configuration Protocol (DHCP) server.

Syntax

To start the **pxed** daemon using the system resource controller:

startsrc -s **pxed** [**-a**]

To start the **pxed** daemon without using the system resource controller:

pxed [**-f** *ConfigurationFile*]

Description

The Preboot Execution Environment defines a protocol and mechanism through which network-connected client systems can automatically download boot images from a network server to start their operating system. As an extension to the BOOTP and DHCP protocols, it provides the configuration ability for administrators that are not necessarily DHCP or network administrators to manage the operating systems installed on the PXE-capable client systems.

Like a DHCP server, the PXE Proxy DHCP server provides information needed by a PXE client to locate and download its appropriate boot files from a network server. However, the PXE Proxy DHCP server does not administer client IP addresses or other DHCP client options.

The PXE Proxy DHCP server is intended to be used when the management of the system boot images must be separated from the management of the DHCP addresses and DHCP client network configurations. The **pxed** daemon can be configured to run on a system that is the DHCP server or is not the DHCP server.

Flags

-a	The argument to be supplied.
-f *ConfigurationFile*	Specifies the path and name of the configuration file that is to be used by the server. If unspecified, the default is **/etc/pxed.cnf**.

Exit Status

This command returns the following exit values:

0	Successful completion.
>0	An error occurred.

Security

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

The PXE protocol allows for a nonprivileged user to be the administrator of the PXE client boot images because the **pxed** daemon listens to client messages on ports other than the well-known, protected DHCP server port. However, to configure such an environment, the DHCP server must be running on the same server system as the **pxed** daemon, and the file permissions on the **pxed** daemon must be changed for non-root execution.

Files

/usr/sbin/pxed	Contains the PXE Proxy DHCP server daemon.
/usr/sbin/db_file.dhcpo	Implements a database to be used by the PXE Proxy DHCP server and the DHCP server to store, retrieve, and manage configuration information.
/etc/pxed.cnf	The default configuration file for the **pxed** daemon.

Related Information

The **dhcpsd** daemon, **binld** daemon.

---

qadm Command

Purpose

Performs system administration functions for the printer spooling system.

Syntax

qadm { **-G** } | { [**-D** *Printer*] [**-K** *Printer*] [**-U** *Printer*] [**-X** *Printer*] }

Description

The **qadm** command is a front-end command to the **enq** command. This command brings printers, queues, and the spooling system up or down and also cancels jobs. The **qadm** command translates the requested flags into a format that can be run by the **enq** command.

The **qadm** command works only on local print jobs. Remote print is not supported.

Note: You must either have root user authority or belong to the printq group to run this command.

You 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 qadm** fast path to run this command.

Flags

-D *Printer*	Brings down the printer you name in the *Printer* variable. The **qdaemon** process stops sending jobs to the device. Entering the **qchk -P** *Printer* command, where *Printer* matches the *Printer* variable in the **-D** flag, reports the device is *down*. The **qadm** command allows current jobs to finish before stopping the printer.
-G	Gracefully brings down the queuing system. This flag temporarily interrupts the **qdaemon** process after all currently running jobs on all queues are finished. Use of this flag is the only way to bring the system down without causing such problems as jobs hanging up in the queue.
-K *Printer*	Brings down the printer you name in the *Printer* variable, ending all current jobs immediately. Jobs remain in the queue and run again when the printer is brought back up.
-U *Printer*	Brings up the printer you name in the *Printer* variable. The **qdaemon** process sends jobs to the printer again. Entering the **qchk -P** *Printer* command, where *Printer* matches the *Printer* variable in the **-U** flag, reports the device is *ready*.
-X *Printer*	Cancels all the jobs of the user that excuted the command. If you have root user privileges or are a member of the printq group, then all jobs on the queue system will be cancelled.

Note: When **-U** and **-D** flags are used together, the **-U** flag has higher priority.

Examples

1. To bring the queuing system down gracefully, enter:

   ```
   qadm -G
   ```

2. To cancel all of a particular user's jobs on printer `lp0`, or all jobs on printer `lp0` if you are have root user authority, enter:

   ```
   qadm -X lp0
   ```

3. To bring up the printer `lpd0` attached to queue `lp0`, enter:

   ```
   qadm -U lp0:lpd0
   ```

Files

/usr/sbin/qdaemon	Contains the **qdaemon** daemon.
/var/spool/lpd/qdir/*	Contains the job description files.
/var/spool/lpd/stat/*	Contains information on the status of the devices.
/var/spool/qdaemon/*	Contains the temporary copies of enqueued files.
/etc/qconfig	Contains the configuration file.
/etc/qconfig.bin	Contains the digested, binary version of the **/etc/qconfig** file.

Related Information

The **enq** command, **qcan** command, **qdaemon** command, **qchk** command, **qpri** command, **qprt** command.

The **/etc/qconfig** file.

Starting and Stopping a Print Queue in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Spooler Overview for System Management in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.2 Web-based System Manager Administration Guide*.

---

qcan Command

Purpose

Cancels a print job.

Syntax

qcan [**-X**] [**-x** *JobNumber*] [**-P** *Printer*]

Description

The **qcan** command cancels either a particular job number or all jobs in a print queue.

You 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 qcan** fast path to run this command.

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 `qcan -x 123`, causes the **qdaemon** to cancel the first matching job number it finds in its internal list, which may be `1123`. By having the additional information that the **-W** flag provides, you can cancel a specific job number.

Flags

-P *Printer*	Specifies the *Printer* where either all jobs or the selected job number will be canceled.
-x *JobNumber*	Specifies that only the job number specified by the *JobNumber* variable be canceled.
-X	Cancels all jobs or all jobs for the specified printer. If you have root user authority, all jobs on that queue are deleted. If you do not have root user authority, only jobs you submitted will be canceled. This flag is only valid for local print jobs.

Examples

1. To cancel all jobs queued on printer `lp0`, enter:

   ```
   qcan  -X  -P lp0
   ```

2. To cancel job number `123` on whatever printer the job is on, enter:

   ```
   qcan  -x 123
   ```

Files

/usr/sbin/qdaemon	Contains the **qdaemon** daemon.
/var/spool/lpd/qdir/*	Contains the job description files.
/var/spool/lpd/stat/*	Contains information on the status of the devices.
/var/spool/qdaemon/*	Contains the temporary copies of enqueued files.
/etc/qconfig	Contains the configuration file.
/etc/qconfig.bin	Contains the digested, binary version of the **/etc/qconfig** file.

Related Information

The **enq** command, **qadm** command, **qchk** command, **qpri** command, **qprt** command.

The **/etc/qconfig** file.

Canceling a Print Job (qcan Command) in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices*.

Spooler Overview for System Management in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.2 Web-based System Manager Administration Guide*.

qchk Command

Purpose

Displays the status of a print queue.

Syntax

qchk [**-A**] [**-L** | **-W**] [**-P** *Printer*] [**-#** *JobNumber*] [**-q**] [**-u** *UserName*] [**-w** *Delay*]

Description

The **qchk** command displays the current status information regarding specified print jobs, print queues, or users. Use the appropriate flag followed by the requested name or number to indicate specific status information. If you run the **qchk** command with no flags, the status of the default queue is returned.

You 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 qchk** fast path to run this command.

Flags

-# *JobNumber*
Requests the status of the job number specified by the *JobNumber* variable. The **qchk** command looks for *JobNumber* on the default queue when the **-#***JobNumber* flag is used alone. To search for *JobNumber* on all queues **-#** flag must be used with the **-A** flag. The **-#** flag may also be used in conjunction with the **-P** *Queue* flag.

Notes:

1. Specify the **-P** *Queue* to override the default destination printer.
2. If jobs 1, 2, and 3 are in the printer queue, and you specify that you want the status of job 3 while job 1 is running, the status information will show job 1 and job 3, not only job 3.
3. If you specify a job number that does not exist, the system displays the current job number on the queue instead of an error message.

-A
Requests the status of all queues.

-L
Specifies information be displayed in long-form mode. This flag cannot be used with the **-W** flag. If the **-L** flag and the **-W** flag are used simultaneously, the first one specified takes precedence.

-P *Printer*
Requests the status of the printer specified by the *Printer* variable.

-q
Requests the status of the default print queue.

-u *UserName*
Requests the status of all print jobs sent by the user specified by the *UserName* variable.

-W
Specifies information be displayed in wide-form mode with longer queue names, device names, and job numbers. Larger job number information is available on AIX 4.3.2 and later. This flag cannot be used with the **-L** flag. If the **-L** flag and the **-W** flag are used simultaneously, the first one specified takes precedence.

-w *Delay*
Updates requested status information at intervals, in seconds, as specified by the *Delay* variable until all print jobs are finished.

Examples

1. To display the default print queue, enter:

```
qchk -q
```

2. To display the long status of all queues until empty, while updating the screen every 5 seconds, enter:

   ```
   qchk  -A  -L  -w 5
   ```

3. To display the status for printer lp0, enter:

   ```
   qchk  -P lp0
   ```

4. To display the status for job number 123, enter:

   ```
   qchk  -# 123
   ```

5. To display the status of all print jobs while restricting the queue status to only printer lp0, enter:

   ```
   qchk  -A  -P lp0
   ```

6. To display the wide status of the default print queue for AIX 4.2.1 or later, enter:

   ```
   qchk  -W  -q
   ```

Files

/usr/sbin/qdaemon	Contains the **qdaemon** daemon.
/var/spool/lpd/qdir/*	Contains the job description files.
/var/spool/lpd/stat/*	Contains information on the status of the devices.
/var/spool/qdaemon/*	Contains the temporary copies of enqueued files.
/etc/qconfig	Contains the configuration file.
/etc/qconfig.bin	Contains the digested, binary version of the **/etc/qconfig** file.

Related Information

Spooler Overview for System Management in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

The **enq** command, **qadm** command, **qcan** command, **qpri** command, **qprt** command.

The **/etc/qconfig** file.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.2 Web-based System Manager Administration Guide*.

Checking Print Job Status (qchk Command) in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

---

qdaemon Command

Purpose

Schedules jobs enqueued by the **enq** command.

Syntax

qdaemon

Description

The **qdaemon** command is a background process (usually started by the **startsrc** command) that schedules printing jobs enqueued by the **enq** command.

The **/etc/qconfig** file should not be edited while there are active jobs in any queue. Editing includes both manual editing and use of the **mkque**, **rmque**, **chque**, **mkquedev**, **rmquedev**, or **chquedev** commands. It

is recommended that all changes to the **/etc/qconfig** file be made using these commands. However, if manual editing is desired, first issue the **enq -G** command to bring the queuing system and the qdaemon to a stop after all jobs are processed. Then edit the **/etc/qconfig** file and restart the qdaemon with the new configuration.

Security

Privilege Control: Only the root user and members of the printq group should have execute (x) access to this command.

Auditing Event	Information
ENQUE_exec	Queue name, job name, host name, file name, user name

Files

/usr/sbin/qdaemon	Contains the **qdaemon** daemon.
/var/spool/lpd/qdir/*	Contains the job description files.
/var/spool/lpd/pio/@local/fullmsg	Contains a flag file whose existence activates **qdaemon** messages to contain complete information.
/var/spool/lpd/stat/*	Contains information on the status of the devices.
/var/spool/qdaemon/*	Contains the temporary copies of enqueued files.
/etc/qconfig	Contains the configuration file.
/etc/qconfig.bin	Contains the digested, binary version of the **/etc/qconfig** file.

Related Information

The **cancel** command, **chque** command, **chquedev** command, **disable** command, **enable** command, **qstatus** command, **lp** command, **lpd** command, **lpq** command, **lpr** command, **lprm** command, **lpstat** command, **lsallq** command, **lsallqdev** command, **lsque** command, **lsquedev** command, **mkque** command, **mkquedev** command, **qprt** command, **qadm** command, **qchk** command, **qcan** command, **qpri** command, **rmque** command, **rmquedev** command, **startsrc** command.

The **/etc/qconfig** file.

Printer Overview for System Management in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Queuing System Overview for System Management in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

National Language Support Overview in *AIX 5L Version 5.2 National Language Support Guide and Reference*.

System Resource Controller Overview in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*.

Understanding the Interaction between qdaemon and the Backend in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

---

qhld Command

Purpose

Holds and releases a spooled print job.

Syntax

qhld [**-r**] { **-#***JobNumber* [**-P***Queue*] | **-P***Queue* | **-u***User* [**-P***Queue*] }

Description

The **qhld** command holds print jobs in a spooled state. The job to be held is designated by job number, queue, or user name. The **-r** flag releases the hold on the print job.

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 hold a specific job.

For example, `qchk` might display job number `123` twice while, `qchk -W` would display job number `1123` and `2123`. If you want to hold job number 2123, specifying `qhld -# 123`, causes the **qdaemon** to hold the first matching job number it finds in its internal list, which may be `1123`. By having the additional information that the **qstatus -W** provides, you can hold a specific job number.

Flags

-#*JobNumber*	Specifies the print job number to be held.
-P*Queue*	Specifies the print queue to be held.
-r	Releases the print job by number, queue, or user name.
-u*User*	Specifies the name of user whose print jobs are to be held.

Examples

1. To hold the print job number `300`, enter:
   ```
   qhld -#300
   ```
2. To hold all print jobs on queue `lp0`, enter:
   ```
   qhld -P lp0
   ```
3. To hold all jobs that belong to user `fred`, enter:
   ```
   qhld -u fred
   ```
4. To release job number `300`, enter:
   ```
   qhld -#300 -r
   ```
5. To release all the jobs on queue `lp0`, enter:
   ```
   qhld -Plp0 -r
   ```
6. To release all jobs that belong to user `fred`, enter:
   ```
   qhld -u fred -r
   ```

Files

/usr/sbin/qdaemon	Contains the **qdaemon** daemon.
/var/spool/lpd/qdir/*	Contains the job description files.
/var/spool/lpd/stat/*	Contains information on the status of the devices.
/var/spool/qdaemon/*	Contains the temporary copies of enqueued files.
/etc/qconfig	Contains the configuration file.
/etc/qconfig.bin	Contains the digested, binary version of the **/etc/qconfig** file.

Related Information

The **qprt** command, **qmov** command, **enq** command,

Printer Overview for System Management in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Spooler Overview for System Management in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

---

qmov Command

Purpose

Moves spooled print jobs to another queue.

Syntax

qmov -m*NewQueue* { **-#***JobNumber* [**-P***Queue*] | **-P***Queue* | **-u***User* [**-P***Queue*] }

Description

The **qmov** command moves spooled print jobs to another print queue. The print job to be moved is identified by job number, queue, or user name. The format of the command requires the queue where the job is to be moved to as the first argument and the name of the job to move as the second argument.

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 move a specific job.

For example, `qchk` might display job number `123` twice while, `qchk -W` would display job number `1123` and `2123`. If you want to move job number 2123, specifying `qmov -# 123`, causes the **qdaemon** to move the first matching job number it finds in its internal list, which may be `1123`. By having the additional information that the **qstatus -W** provides, you can move a specific job number.

Flags

-#*JobNumber*	Specifies the job number of the print job to be moved.
-m*NewQueue*	Specifies the name of the destination print queue.
-P*Queue*	Specifies the present print queue of the job to be moved.
-u*User*	Specifies the name of the user whose print jobs are to be moved.

Examples

1. To move job number `280` to queue `lp0`, enter:
   ```
   qmov -mlp0 -#280
   ```
2. To move all print jobs on queue `lp1` to queue `lp0`, enter:
   ```
   qmov -mlp0 -Plp1
   ```
3. To move all of Mary's print jobs to queue `lp0`, enter:
   ```
   qmov -mlp0 -u mary
   ```

Files

/usr/sbin/qdaemon	Contains the **qdaemon** daemon.

/var/spool/lpd/qdir/*	Contains the job description files.
/var/spool/lpd/stat/*	Contains information on the status of the devices.
/var/spool/qdaemon/*	Contains the temporary copies of enqueued files.
/etc/qconfig	Contains the configuration file.
/etc/qconfig.bin	Contains the digested, binary version of the **/etc/qconfig** file.

Related Information

The **enq** command, **qhld** command, **qprt** command.

Printer Overview for System Management in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Spooler Overview for System Management in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

---

qosadd Command

Purpose

Adds a QoS (Quality of Service) Service Category or Policy Rule.

Syntax

To add a Service Category:

qosadd [**-s** *ServiceCategory*] [**-t** *OutgoingTOS*] [**-b** *MaxTokenBucket*] [**-f** *FlowServiceType*] [**-m** *MaxRate*] **service**

To add a Policy Rule:

qosadd [**-s** *ServiceCategory*] [**-r** *ServicePolicyRules*] [**-l** *PolicyRulePriority*] [**-n** *ProtocolNumber*] [**-A** *SrcAddrRange*] [**-a** *DestAddrRange*] [**-P** *SrcPortRange*] [**-p** *DestPortRange*] **policy**

Description

The **qosadd** command adds the specified Service Category or Policy Rule entry in the **policyd.conf** file and installs the changes in the QoS Manager.

Flags

Flags with service add:

-s	The name of the **ServiceCategory** attribute, which is mandatory.
-t	The **OutgoingTOS** attribute, specified as an 8 bit binary number.
-b	The **MaxTokenBucket** attribute, specified in Kb (Kilobits).
-f	The **FlowServiceType** attribute, which is ControlledLoad or Guaranteed.
-m	The **MaxRate** attribute, which is specified in Kbps (Kilobits per second).

Flags with policy add:

-s	The name of the **ServiceCategory** attribute, which is mandatory.
-r	The name of the **ServicePolicyRules** attribute, which is mandatory.
-l	The **PolicyRulePriority** attribute, which is a positive integer.
-n	The **ProtocolNumber** attribute, which is defined in the **/etc/protocols** file.
-A	The **SrcAddrRange** attribute, which is the Source IP address range from a1 to a2 where a2 >= a1.

-a	The **DestAddrRange** attribute, which is the Destination IP address range from i1 to i2 where i2 >= i1.
-P	The **SrcPortRange** attribute, which is the Source Port range from a1 to a2 where a2 >= a1.
-p	The **DestPortRange** attribute, which is the Destination Port range from i1 to i2 where i2 >= i1.

Exit Status

0	Successful completion
Positive Integer	An error occurred.

Examples

1. To add the sc01 service, type:

   ```
   qosadd -s sc01 -t 10000001 -b 81 -f ControlledLoad -m 41 service
   ```

2. To add the pr01 policy, type:

   ```
   qosadd -s sc01 -r pr01 -l 2 -n 17 -A 9.3.25.1-9.3.25.10 -a  9.3.25.33-9.3.25.33 -p 9001-9010 -P 9000-9000 policy
   ```

3. To add the sc02 service, type:

   ```
   qosadd -s sc02 -t 10000001 -b 81 service
   ```

4. To add the pr02 policy, type:

   ```
   qosadd -s sc02 -r pr02 -l 2 -n 17 policy
   ```

Related Information

The **qosstat** command, **qosmod** command, **qosremove** command, **qoslist** command.

---

qoslist Command

Purpose

Lists a specific QoS (Quality of Service) Service Category or Policy Rule or lists all of them.

Syntax

To list a Service Category:

qoslist [*ServiceCategory*] **service**

To list a Policy Rule:

qoslist [*ServicePolicyRule*] **policy**

Description

The **qoslist** command lists the specified Service Category or Policy Rule. The **qoslist** command lists all Service Categories or Policy Rules if no specific name is given.

Exit Status

0	Successful completion
Positive Integer	An error occurred.

Examples

1. To list the sc01 service, type:
   ```
   qoslist sc01 service
   ```
2. To list the the pr01 policy, type:
   ```
   qoslist pr01 policy
   ```
3. To list all of the QoS ServiceCategories, type:
   ```
   qoslist service
   ```
4. To list all of the QoS PolicyRules, type:
   ```
   qoslist policy
   ```

Related Information

The **qosstat** command, **qosmod** command, **qosremove** command, **qosadd** command.

---

qosmod Command

Purpose

Modifies an existing QoS (Quality of Service) Service Category or Policy Rule.

Syntax

To modify an existing Service Category:

qosmod [**-s** *ServiceCategory*] [**-t** *OutgoingTOS*] [**-b** *MaxTokenBucket*] [**-f** *FlowServiceType*] [**-m** *MaxRate*] **service**

To modify an exisiting Policy Rule:

qosmod [**-s** *ServiceCategory*] [**-r** *ServicePolicyRules*] [**-l** *PolicyRulePriority*] [**-n** *ProtocolNumber*] [**-A** *SrcAddrRange*] [**-a** *DestAddrRange*] [**-P** *SrcPortRange*] [**-p** *DestPortRange*] **policy**

Description

The **qosmod** command modifies the specified Service Category or Policy Rule entry in the **policyd.conf** file and installs the changes in the QoS Manager.

The **qosmod** command clears out all the statistics of the old policy. When a **qosstat** command is executed immediately after **qosmod**, the user may not see all the data connections that were using the older rule shifted to the modified rule. This is because the reclassification of the data connection is delayed until a data packet arrives on that connection.

Note: Modifying the priority or filter spec of the rule only results in reclassification of the data connections which use that particular rule. Connections using other rules maintain their existing classification.

Flags

Flags with service modify:

-s	The name of the **ServiceCategory** attribute, which is mandatory.
-t	The **OutgoingTOS** attribute, specified as an 8-bit binary number.
-b	The **MaxTokenBucket** attribute, specified in Kb (Kilobits).
-f	The **FlowServiceType** attribute, which is ControlledLoad or Guaranteed.
-m	The **MaxRate** attribute, which is specified in Kbps (Kilobits per second).

Flags with policy modify:

-s	The name of the **ServiceCategory** attribute, which is mandatory.
-r	The name of the **ServicePolicyRules** attribute, which is mandatory.
-l	The **PolicyRulePriority** attribute, which is a positive integer.
-n	The **ProtocolNumber** attribute, which is defined in the **/etc/protocols** file.
-A	The **SrcAddrRange** attribute, which is the Source IP address range from a1 to a2, where a2 >= a1.
-a	The **DestAddrRange** attribute, which is the Destination IP address range from i1 to i2, where i2 >= i1.
-P	The **SrcPortRange** attribute, which is the Source Port range from a1 to a2, where a2 >= a1.
-p	The **DestPortRange** attribute, which is the Destination Port range from i1 to i2, where i2 >= i1.

Exit Status

0	Successful completion
Positive Integer	An error occurred.

Examples

1. To modify the sc01 service, type:

   ```
   qosmod -s sc01 -t 10001100 -b 84 -f Guaranteed  service
   ```

2. To modify the pr01 policy, type:

   ```
   qos -s sc01 -r pr01 -l 10 -n 6 -A 9.3.25.15-9.3.25.20 -a 9.3.25.39-9.3.25.39 -p 9015-9020 policy
   ```

3. To modify the sc02 service, type:

   ```
   qosmod -s sc02 -t 10001111 service
   ```

4. To modify the pr02 policy, type:

   ```
   qosmod -s sc02 -r pr02 -l 13 -n 6 policy
   ```

Related Information

The **qosstat** command, **qoslist** command, **qosremove** command, **qosadd** command.

---

qosremove Command

Purpose

Removes a QoS (Quality of Service) Service Category or Policy Rule.

Syntax

To remove a Service Category:

qosremove [*ServiceCategory*] **service**

To remove a Policy Rule:

qosremove [*ServicePolicyRule*] **policy**

To remove all the Policies and Service categories installed in the kernel:

qosremove all

Description

The **qosremove** command removes the specified Service Category or Policy Rule entry in the **policyd.conf** file and the associated policy or service in the QoS Manager.

Exit Status

0	Successful completion
Positive Integer	An error occurred.

Examples

1. To remove the sc01 service, type:

   ```
   qosremove sc01 service
   ```

2. To remove the pr01 policy, type:

   ```
   qosremove pr01 policy
   ```

Related Information

The **qosstat** command, **qosmod** command, **qoslist** command, **qosadd** command.

---

qosstat Command

Purpose

Show Quality of Service (QoS) status.

Syntax

qosstat [**-A**] [**-n**] [**-a**]

Description

The **qosstat** command displays information about the installed Qos policies.

Flags

-A	Displays the addresses of the QoS control block for each policy.
-n	Displays network addresses and ports as numbers. Without this flag, **qosstat** translates numeric addresses and ports and displays them symbolically.
-a	Displays all policies. Without this flag, only policy conditions with associated connections are displayed.

Example

```
qosstat -na

Action:
  Token bucket rate (B/sec): 10240
  Token bucket depth (B): 1024
  Peak rate (B/sec): 10240
  Min policied unit (B): 20
  Max packet size (B): 1452
  Type: IS-CL
  Flags: 0x00001001 (POLICE,SHAPE)

  Statistics:
    Compliant packets: 1423 (440538 bytes)
```

```
  Conditions:
    Source address        Dest address          Protocol
    192.168.127.39:8000   192.168.256.29:35049  tcp       (1 connection)

Action:
  Token bucket rate (B/sec): 10240
  Token bucket depth (B): 1024
  Peak rate (B/sec): 10240
  Outgoing TOS (compliant): 0xc0
  Outgoing TOS (non-compliant): 0x00
  Flags: 0x00001011 (POLICE,MARK)
  Type: DS

  Statistics:
    Compliant packets: 335172 (20721355 bytes)
    Non-compliant packets: 5629 (187719 bytes)

  Conditions:
    Source address        Dest address          Protocol
    192.168.127.39:80     *:*                   tcp       (1 connection)
    192.168.127.40:80     *:*                   tcp       (5 connections)
```

Related Information

TCP/IP Quality of Service (QoS) in the *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

---

qpri Command

Purpose

Prioritizes a job in the print queue.

Syntax

qpri -# *JobNumber* **-a** *PriorityNumber*

Description

The **qpri** command prioritizes a job in a print queue by specifying the job number and giving it a priority number.

The **qpri** command works only on local print jobs and the local side of remote queues. Remote print jobs are not supported. Also, you must have root user authority or belong to the printq group to run this command.

You 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 qpri** fast path to run this command.

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 alter the priority of a specific job.

For example, `qchk` might display job number `123` twice while, `qchk -W` would display job number `1123` and `2123`. If you want to alter the priority of job number 2123, specifying `qpri -# 123`, causes the **qdaemon** to

alter the priority of the first matching job number it finds in its internal list, which may be 1123. By having the additional information that the **qstatus -W** provides, you can alter the priority of a specific job number.

Flags

-# *JobNumber*	Specifies the job number on which to change priority.
-a *PriorityNumber*	Specifies the new priority number for the print job specified by the *JobNumber* variable. The range of priority numbers is 1 through 20, except for the root user or a member of the printq group, who can select priority numbers from 1 through 30.

Example

To change job number 123 to priority number 18, enter:

```
qpri  -# 123  -a 18
```

Files

/usr/sbin/qdaemon	Contains the **qdaemon** daemon.
/var/spool/lpd/qdir	Contains the job description files.
/var/spool/lpd/stat	Contains information on the status of the devices.
/var/spool/qdaemon	Contains the temporary copies of enqueued files.
/etc/qconfig	Contains the configuration file.
/etc/qconfig.bin	Contains the digested, binary version of the **/etc/qconfig** file.

Related Information

The **enq** command, **qadm** command, **qcan** command, **qchk** command, **qprt** command.

The **/etc/qconfig** file.

Prioritizing a Print Job (qpri Command) in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices*.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.2 Web-based System Manager Administration Guide*.

---

qprt Command

Purpose

Starts a print job.

Syntax

qprt [**-a** *PreviewOption*] [**-A** *Level*] [**-b** *BottomMargin*] [**-B** *Value*] [**-c**] [**-C**] [**-d** *InputDataType*] [**-D** *"User"*] [**-e** *EmphasizedOpt*] [**-E** *DblHigh*] [**-f** *Filter*] [**-F** *Name*] [**-g** *Begin*] [**-G** *Coord*] [**-h** *"Header"*] [**-H** *"HostName"*] [**-i** *Indent*] [**-I** *FontPath*] [**-j** *Init*] [**-J** *Restore*] [**-k** *Color*] [**-K** *Condense*] [**-l** *Length*] [**-L** *LineWrap*] [**-m** *Message*] [**-M** *MessageFile*] [**-n**] [**-N** *NumberCopies*] [**-O** *PaperHand*] [**-p** *Pitch*] [**-P** *Queue* [**:***QueueDevice*]] [**-Q** *Value*] [**-q** *Quality*] [**-r**] [**-R** *Priority*] [**-s** *NameType*] [**-S** *Speed*] [**-t** *TopMargin*] [**-T** *"Title"*] [**-u** *PaperSrc*] [**-U** *Directional*] [**-v** *LinesPerIn*] [**-V** *Vertical*] [**-w** *PageWidth*] [**-W** *DblWide*] [**-x** *LineFeed*] [**-X** *CodePage*] [**-y** *DblStrike*] [**-Y** *Duplex*] [**-z** *Rotate*] [**-Z** *FormFeed*] [**-#** { **j** | **H** | **v** }] [**-=** *OutputBin*]{ *File* | **-** } ...

Description

The **qprt** command creates and queues a print job to print the file specified by the *File* parameter. To print a file from standard input, specify a **-** (dash) instead of a file name. If you specify multiple files, then they all together make up one print job. The **qprt** command prints the files in the order you specify them.

To print a file, you must have read access to it. Using the **-r** flag you can remove a file after printing it. To remove a file, you must have write access to the directory that contains it. If you want the **qprt** command to notify you when a print job completes, specify the **-n** flag.

You can use the **-B** flag in conjunction with the **-D**, **-H**, and **-T** flags to customize burst pages. Burst pages mark the beginning, end, or both of a print job. To mark the beginning and end of a print job with burst pages, use the **-B aa** flag.

All flags are optional and you can specify them in any order. The **qprt** command ignores spaces between a flag and its argument. You can group flags without arguments after a single **-** (dash). All flags and their arguments must precede the *File* parameter.

You 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 qprt** fast path to run this command.

Some of the flags and arguments listed in this command are invalid for particular printer types. If you experience problems using an option, you can use SMIT to preview a **qprt** command statement. See the ″System Management Interface Tool (SMIT): Overview″ in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices*. Alternatively, consult your printer manual to find out what options your printer supports.

Flags

-#{jlhlv} Specifies a special functionality. The possible values for the *Value* variable are:

- **j** Displays a job number for the specified print job.
- **H** Queues the print job but holds it in a HELD state.
- **v** Validates the specified printer backend flag values. As part of the validation process, the command performs legality checking for illegal flag values, type checking, range checking, list checking, and other types of validation. Typically, the validation of backend flag values is useful because illegal flags are identified when the print job is submitted rather than at a later stage when the print job is processed.

-= *OutputBin* Specifies the output bin destination for a print job. If you do not specify this flag, it uses the default value from the printer driver.

The possible values for *OutputBin* are:

- **0** Top printer bin.
- **1 - 49** High Capacity Output (HCO) bins 1 - 49.
- **>49** Printer specific output bins.

Note: Valid output bins are printer dependent.

-a *PreviewOption*

Previews parameter values for a print job without actually printing any files. You can specify a **0** or a **1** for the *PreviewOption* variable. If you specify **0**, the **qprt** command preview displays normal print processing. If you specify a **1**, the command returns a list of the flag values and the filter pipeline that would be used to convert the input data type to the data type expected by the printer. These flag values are the default command line flag values from the configuration database, overridden by any flag parameters specified on the command line.

Only flags that are valid for the *InputDataType* variable specified (or defaulted) for the **-d** flag are shown. Flag values related only to the spooling of your print job, instead of the actual printing, are not shown. The default values for the spooling flags are included with the flag descriptions. The flag values are not checked to verify that they are supported by the printer.

The pipeline of filters shows the filter commands (and the flag values passed to the filter commands) that would process the data from your print files before it is sent to the printer. You can review the description for each of the filter commands to determine the type of filtering that is performed.

-A *Level*

Sets the level of diagnostic output. Diagnostic output is useful for diagnosing errors encountered by a filter pipeline that is processing a print file, a header page, or a trailer page. Diagnostic output is mailed to the user who submitted the print job. You can specify one of the following levels:

0 Discards any standard error output produced.

1 Returns flag values, the standard error output, and the complete pipeline that produced any standard error output.

2 Returns the flag values, standard error output (if any), and complete pipelines, regardless of whether an error is detected. If an error is detected, the print job is terminated.

3 Similar to a value of **2**, except that the file is not printed.

A value of **1** is recommended. A value of **0** is useful if a filter in a pipeline produces output to standard error, even if no error is encountered (for example, status information). A value of **2** or **3** is useful for diagnosing a problem, even if the problem does not cause any output to standard error.

-b *BottomMargin*

Specifies the bottom margin, the number of blank lines to be left at the bottom of each page.

-B *Value*

Prints burst pages. The *Value* variable consists of a two-character string. The first character applies to header pages. The second character applies to trailer pages. The following values are valid:

a Always print the (header or trailer) page for each file in each print job.

n Never print the (header or trailer) page.

g Print the (header or trailer) page once for each print job (group of files).

For example, the **-B ga** flag prints a header page at the beginning of each print job and a trailer page after each file in each print job.

> **Note:** In a remote print environment, the default is determined by the remote queue on the server.

-c

Copies each print file and prints from the copy. Specify this flag if you plan to modify the print file or files after the **qprt** command is issued, but before the print job completes.

If this flag is not specified and the print job is printed on the same node where it was submitted, copies of the print file or files are not made. Printing occurs directly from the file or files you specified with the *File* parameter.

-C Mails messages generated by your print job to you, even if you are logged in. By default, the **qprt** command displays messages on the console.

The **-C** flag only applies to local print jobs. If you want to be notified when a job sent to a remote printer is completed, use the **-n** flag to receive a mail message.

> **Note:** You cannot redirect certain messages from the **qdaemon** and the printer backend in any way. They are sent directly to the **/dev/console** file.

-d *InputDataType* Identifies the input data type of the file or files to print. Based on the input data type and the data type expected by the printer, the print files are passed through filters (if necessary) before being sent to the printer. You can specify any of the following input data types:

a	Extended ASCII
c	PCL
d	Diablo 630
g	Hewlett-Packard GL
p	Pass-through (sent to printer unmodified)
s	PostScript

If the printer you select does not support the specified input data type, and if filters are not available to convert the data type of your print file or files to a data type supported by the printer, the print job terminates with an error message.

-D *"User"* Labels the output for delivery to *User*. Normally the output is labeled for delivery to the user name of the person issuing the **qprt** command request. The value of *User* must be a single word meeting the same requirements of a regular user ID.

-e *EmphasizedOpt* Sets emphasized print to one of the following:

+	Use emphasized print.
!	Do not use emphasized print.

-E *DblHigh* Sets double-high print to one of the following:

+	Use double-high print.
!	Do not use double-high print.

-f *Filter* Identifies the filter to pass your print files through before sending them to the printer. The identifiers are similar to the filter flags available with the **lpr** command. The available filter identifiers are **p**, which invokes the **pr** filter, and **n**, which processes output from the **troff** command.

-F *Name* Specifies the list of X font files containing the image of characters to be used for printing. Items in the list must be separated by commas. The *Name* parameter value can be full path names, font alias names, or XLFD names. The **-F** Name flag is effective only for MBCS printer queues.

-g *Begin* Sets the page number to begin printing. This flag is recognized only if the print files are to be formatted (for example, with the **-d a** flag). It is not recognized for pass-through (the **-d p** flag), PostScript (the **-d s** flag), and other types of data that are already formatted.

-G *Coord* Indicates how to print pages on laser printers that cannot print to the edge of the paper. Use one of the following for the *Coordinate* variable:

+	Whole page coordinate system
!	Print page coordinate system

-h *"Header"*
Specifies the header text for use by the **pr** command when the **-f p** flag is also specified. If this flag is not specified, the **pr** command uses the print file name as the header.

This flag is useful if you also specified the **-c** flag. With the **-c** flag, the print file name used by the **pr** command as the default header is the name of a temporary file generated by the spooler, instead of the file name you specified with the **qprt** command.

-H *"HostName"*
Sets the host name on the header page.

-i *Indent*
Indents each line the specified number of spaces. You must include the *Indent* variable in the page width specified by the **-w** flag.

-I *FontID*
(uppercase i) Specifies a font identifier. Specifying a font identifier overrides the pitch (the **-p** flag) and type style (the **-s** flag). The **-I***FontID* command is effective for single byte code set print queues only.

-I*FontPath*
(uppercase i) Specifies the comma-separated list of font paths required for the **-F** flag when the font files are designated with a font alias name or an XLFD name. The *FontPath* flag is effective only for MBCS printer queues.

-j *Init*
Initializes the printer before each file is printed. You can specify any of the following:

0 No initialization

1 Full initialization

2 Emulator selection only

-J *Restore*
Restores the printer at the end of the print job. You can specify one of the following:

+ Restore at the end of the print job.

! Do not restore at the end of the print job.

-k *Color*
Specifies the print color. Typical values are black, red, blue, green, and so on. Consult your printer manual for colors supported and the ribbon position assigned to a particular color.

-K *Condense*
Sets condensed print to one of the following:

+ Use condensed print.

! Do not use condensed print.

-l *Length*
(lowercase L) Sets the page length. If the *Length* variable is 0, page length is ignored, and the output treated as one continuous page. The page length includes the top and bottom margins and indicates the printable length of the paper.

-L *LineWrap*
Sets line wrap for lines wider than the page width to one of the following:

+ Wrap long lines to the next line.

! Truncate long lines at the right margin.

-m *"Message"*
Displays the specified message on the console when the print job is assigned a printer and is ready to begin printing. The print job does not proceed until the message is acknowledged at the console.

-M *MessageFile*
Identifies a file containing message text. This text is displayed on the console when the print job is assigned a printer and is ready to begin printing. The print job does not proceed until the message is acknowledged at the console.

-n
Notifies you when the print job completes. If the **-D** *"User"* flag is also specified, the specified user is notified as well. By default, you are not notified when the print job completes.

-N *NumberCopies*
Specifies the number of copies to print. If this flag is not specified, one copy is printed.

-O *PaperHand*
Sets the type of input paper handling to one of the following:

1 Manual (insert one sheet at a time)

2 Continuous forms

3 Sheet feed

-p *Pitch*
Sets the number of characters per inch. Typical values for *Pitch* are 10 and 12. The actual pitch of the characters printed is also affected by the values for the **-K** (condensed) flag and the **-W** (double-wide) flag.

If you are printing an ASCII file on a PostScript printer, this flag determines the character point size. You can specify positive numbers greater than or equal to 1.

-P *Queue*[:*QueueDevice*]
Specifies the print queue name and the optional queue device name. If this flag is not specified, the following conditions occur:

- If the **LPDEST** environment variable is set, the **qprt** command uses the queue name specified by the **LPDEST** variable. If set, this value is always used, even if the **PRINTER** variable is also set.
- If the **PRINTER** variable is set and no **LPDEST** variable is set, the **qprt** command uses the queue name specified by the **PRINTER** environment variable. Any destination command-line options override both the **LPDEST** and **PRINTER** environment variables.
- If neither the **LPDEST** nor the **PRINTER** variable is set, the **qprt** command uses the system default queue name. (The system default queue name is the name of the first queue defined in the **/etc/qconfig** file.) If the *QueueDevice* variable is not specified, the first available printer configured for the queue is used.

  Note: If multiple printers are configured for the same print queue and one or more of the printers is not suitable for printing your files, you should use the *QueueDevice* variable. Otherwise, the spooler assigns the first available printer.

-q *Quality*
Sets the print quality to one of the following:

0 Fast font

1 Draft quality

2 Near letter quality

3 Enhanced quality

300 300 dots per inch (dpi)

600 600 dpi

-Q *Value*
Sets the paper size. The *Value* for paper size is printer-dependent. Typical values are: **1** for letter-size paper, **2** for legal, and so on. Consult your printer manual for the values assigned to specific paper sizes.

-r
Removes the print files after the print job completes. If this flag is not specified, the print files are not removed.

-R *Priority*
Sets the priority for the print job. Higher values for the *Priority* variable indicate a higher priority for the print job. The default priority value is **15**. The maximum priority value is **20** for most users and **30** for users with root user privilege and members of the system group (group 0).

Note: You cannot use this flag when requesting remote print jobs.

-s *NameType*
Specifies a type style with the *NameType* variable. Examples are courier and prestige. The particular type style choices differ depending on the printer type.

-S *Speed*
Sets high-speed printing to one of the following:

+ Use high-speed printing.

! Do not use high-speed printing.

-t *TopMargin*
Sets the top margin, the number of blank lines left at the top of each page.

-T *"Title"*
Specifies a print job title with the *Text* variable. If this flag is not specified, the first file name on the **qprt** command line is used as the print job title. The print job title is displayed on the header page and on responses to inquiries about queue status.

-u *PaperSrc*
Sets the paper source to one of the following:

1 Primary

2 Alternate

3 Envelopes

-U *Directional*
Sets unidirectional printing to one of the following:

+ Use unidirectional printing.

! Do not use unidirectional printing.

-v *LinesPerIn*
Sets the line density to a number of lines per inch. Typical values for the *LinesPerIn* variable are **6** and **8**.

-V *Vertical*
Sets vertical printing to one of the following:

+ Use vertical printing.

! Do not use vertical printing.

-w *PageWidth*
Sets the page width in number of characters. The page width must include the number of indention spaces specified with the **-i** flag.

-W *DblWide*
Sets double-wide print to one of the following:

+ Use double-wide print.

! Do not use double-wide print.

-x *LineFeed*
Specifies automatic line feed or automatic carriage return:

0 Do not change line feeds, vertical tabs, and carriage returns.

1 Add a line feed for each carriage return.

2 Add a carriage return for each line feed and each vertical tab.

-X *CodePage*
Provides the code page name. Valid values for the *CodePage* variable are ISO8859-1 through ISO8859-9, IBM-943, IBM-eucJP, IBM-eucKR, and IBM-eucTW. The code page in the user's locale definition is the default.

-y *DblStrike*
Sets double-strike print to one of the following:

+ Use double-strike print.

! Do not use double-strike print.

-Y *Duplex*
Sets duplexed output. Duplexed output uses both the front and back of each sheet of paper for printing. You can set one of the following:

0 Simplex

1 Duplex, long edge binding

2 Duplex, short edge binding

-z *Rotate*
Rotates page printer output the number of quarter-turns clockwise as specified by the *Value* variable. The length (**-l**) and width (**-w**) values are automatically adjusted accordingly.

0 Portrait

1 Landscape right

2 Portrait upside-down

3 Landscape left

-Z *FormFeed* Sends a form feed to the printer after each print file. You can specify either of the following:

+ Send a form feed command.

! Do not send a form feed command to the printer. Use this option carefully since it can result in the next print job beginning on the last output page generated by this print job. Printers printing on continuous forms cannot determine the top of the form for subsequent pages.

Examples

1. To queue the `myfile` file to print on the first available printer configured for the default print queue using the default values, enter:

   ```
   qprt myfile
   ```

2. To queue a file on a specific queue, to print the file without using nondefault flag values, and to validate the flag values at the time of the print job submission, enter:

   ```
   qprt  -f p  -e +  -P fastest  -r  -n  -C  -#v somefile
   ```

   This command line passes the `somefile` file through the **pr** command (the **-f p** flag) and prints it using emphasized mode (the **-e +** flag) on the first available printer configured for the queue named **fastest** (the **-P fastest** flag). The **-#v** flag verifies that all flags associated with this command are valid before passing the print job to the printer backend. After the file is printed, it is removed (the **-r** flag), and the user who submitted the print job is notified (the **-n** flag) by mail (the **-C** flag) that the print job completed.

3. To print `myfile` on legal size paper, enter:

   ```
   qprt  -Q2 myfile
   ```

4. To enqueue the `myfile` file and return the job number, enter:

   ```
   qprt -#j myfile
   ```

5. To queue `MyFile` and hold it, enter:

   ```
   qprt -#H MyFile
   ```

Files

/etc/qconfig	Contains the queue and queue device configuration file.
/usr/bin/qprt	Contains the **qprt** command.

Related Information

The **enq** command, **lp** command, **lpr** command,**pr** command, **qadm** command, **qcan** command, **qchk** command, **qpri** command,**smit** command, **troff** command.

The **/etc/qconfig** file.

Printer Overview for System Management, Spooler Overview for System Management, Printer Specific Information, Virtual Printer Definitions and Attributes, Initial Printer Configuration, Starting a Print Job (qprt Command) in *AIX 5L Version 5.2 Guide to Printers and Printing*.

How to Print from a Remote System and How to Print to a Remote System in *AIX 5L Version 5.2 System User's Guide: Communications and Networks*.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.2 Web-based System Manager Administration Guide*.

qstatus Command

Purpose

Provides printer status for the print spooling system.

Syntax

qstatus [**-#** *JobNumber*] [**-A**] [**-L** | **-W**] [**-P** *Printer*] [**-e**] [**-q**] [**-u** *UserName*] [**-w** *DelaySeconds*]

Description

The **qstatus** command performs the actual status function for the print-spooling system. This command is never entered on the command line; it is called by the **enq** command. The **qstatus** command generates status information on specified jobs, printers, queues, or users.

The display generated by the **qstatus** command contains two entries for remote queues. The first entry contains the client's local queue and local device name and its status information. The second entry follows immediately; it contains the client's local queue name (again), followed by the remote queue name. Any jobs submitted to a remote queue are displayed first on the local side and are moved to the remote device as the job is processed on the remote machine.

Since the status commands communicate with remote machines, the status display may occasionally appear to hang while waiting for a response from the remote machine. The command will eventually time-out if a connection cannot be established between the two machines.

Flags

All flags are optional. If flags are not specified, the **qstatus** command returns the status of the following:

- The printer specified by the **LPDEST** variable, if the **LPDEST** environment variable is set. If set, this value is always used, even if the **PRINTER** variable is also set.
- The printer specified by the **PRINTER** environment variable, if the **PRINTER** variable is set and no **LPDEST** variable is set.
- The default printer, if neither the **LPDEST** nor the **PRINTER** variable is set.

> **Note:** Any destination command line options override both the **LPDEST** and the **PRINTER** environment variables.

-# *JobNumber*
Displays current status information for the job specified by the *JobNumber* variable. Normally, the status of all queued jobs is displayed.

1. Specify the **-P** *Queue* to override the default destination printer.
2. If jobs 1, 2, and 3 are in the printer queue, and you specify that you want the status of job 3 while job 1 is running, the status information will show job 1 and job 3, not only job 3.
3. If you specify a job number that does not exist, the system displays the current job number on the queue instead of an error message.

-A
Displays status information on all queues defined in the **/etc/qconfig** file.

-e
Excludes status information from queues that are not under the control of the **qdaemon** command. The status from such queues may appear in different formats. The **-e** flag can be used with any combination of flags.

-L
Displays a long, detailed version of status information as requested on the command line. This flag cannot be used with the **-W** flag. If the **-L** flag and the **-W** flag are used simultaneously, the first one specified takes precedence.

-P *Printer*
Displays current status information for the printer specified by the *Printer* variable. Normally, the default printer is used, or the value of either the **LPDEST** or **PRINTER** environment variable is used. The **LPDEST** variable always takes precedence over the **PRINTER** variable.

-q	Displays the current status of the default queue. The default queue is specified by the **LPDEST** variable, or if a **LPDEST** value does not exist, by the **PRINTER** environment variable. If neither variable exists, the **qstatus** command uses the first queue listed in the **/etc/qconfig** file.
-u *UserName*	Displays current status information for all jobs submitted by the user specified by the *UserName* variable. Normally, the status of all queued jobs is displayed.
-W	Displays a wide version of status information with longer queue names, device names, and job numbers. Longer job number information is available on AIX 4.3.2 and later. This flag cannot be used with the **-L** flag. If the **-L** flag and the **-W** flag are used simultaneously, the first one specified takes precedence.
-w *DelaySeconds*	Displays requested queue information at intervals specified by the *DelaySeconds* variable. When the queue is empty, the display ends.

Examples

1. To display the default print queue, enter:

   ```
   qstatus -q
   ```

2. To display the long status of all queues until empty, while updating the screen every 5 seconds, enter:

   ```
   qstatus -A -L -w 5
   ```

3. To display the status for printer lp0, enter:

   ```
   qstatus -P lp0
   ```

4. To display the status for job number 123, enter:

   ```
   qstatus -# 123 -P lp0
   ```

5. To display the status of all queues in wide format for AIX 4.2.1 or later, enter:

   ```
   qstatus -A -W
   ```

Files

/var/spool/lpd/qdir/*	Contains the job description files.
/etc/qconfig	Contains the configuration file.
/etc/qconfig.bin	Contains the digested, binary version of the **/etc/qconfig** file.
/usr/lib/lpd/rembak	Contains the remote back end.
/usr/lib/lpd/qstatus	Contains the command file.
/var/spool/lpd/stat/*	Contains the status files for the **qstatus** command.

Related Information

The **enq** command, **lpd** command, **lpr** command, **qdaemon** command, **qchk** command, **rembak** command.

The **/etc/qconfig** file.

Spooler Overview for System Management in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Remote Printing Overview in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

quiz Command

Purpose

Tests your knowledge.

Syntax

quiz { **-i** *File* | **-t** | *Category1 Category2* }

Description

The **quiz** command gives associative knowledge tests on various selectable subjects. It asks about items chosen from *Category1* and expects answers from *Category2*. If you do not specify the categories, the **quiz** command lists the available categories, provides instructions, and returns to the shell prompt.

The game provides the correct answer whenever you press the Enter key. When questions run out or when you press the Interrupt (Ctrl-C) or End Of File (Ctrl-D) key sequences, the game reports a score and ends.

Flags

-i *File* Substitutes the named *File* for the standard index file.

Note: In the following syntax description, brackets are normally used to indicate that an item is optional. However, a bold-faced bracket or brace should be entered as a literal part of the syntax. A vertical list of items indicates that one must be chosen. The lines in *File* must have the following syntax:

```
line       = category [:category] . . .
category   = alternate [ |alternate ] . . .
alternate  = [primary]
primary    = character
             [category]
             option
option     = {category}
```

In an index file, the first category of each line must specify the name of an information file. The information file contains the names of files with quiz material. The remaining categories specify the order and contents of the data in each line of the information file. The quiz data in an information file follows the same syntax.

A \ (backslash) is an escape character that allows you to quote syntactically significant characters or to insert a new-line character (*n*) into a line. When either a question or its answer is blank, the **quiz** command does not ask the question. The construct **alab** does not work in an information file. Use **a{b}**.

-t Provides a tutorial. Repeats missed questions and introduces new material gradually.

Examples

1. To start a Latin-to-English quiz, enter:

   ```
   /usr/games/quiz latin english
   ```

   The game displays Latin words and waits for you to enter what they mean in English.
2. To start an English-to-Latin quiz, enter:

   ```
   /usr/games/quiz english latin
   ```
3. To set up a Latin-English quiz, add the following line to the index file:

   ```
   /usr/games/lib/quiz/latin:latin:english
   ```

This line specifies that the **/usr/games/lib/quiz/latin** file contains information about the categories Latin and English.

You can add new categories to the standard index file, **/usr/games/lib/quiz/index**, or to an index file of your own. If you create your own index file, run the **quiz** command with the **-i***File* flag and enter your list of quiz topics.

4. The following is a sample information file:

```
cor:heart
sacerdos:priest{ess}
quando:when|since|because
optat:{{s}he |it }[desires|wishes]\|
desire|wish
alb[us|a|um]:white
```

This information file contains Latin and English words. The : (colon) separates each Latin word from its English equivalent. Items enclosed in { } (braces) are optional. A | (vertical bar) separates two items when entering either is correct. The [ ] (brackets) group items separated by vertical bars.

The first line accepts only the answer `heart` in response to the Latin word `cor`. The second accepts either `priest` or `priestess` in response to `sacerdos`. The third line accepts `when`, `since`, or `because` for `quando`.

The \ (backslash) at the end of the fourth line indicates that this entry continues on the next line. In other words, the fourth and fifth lines together form one entry. This entry accepts any of the following in response to `optat`:

```
she desires it desires desire
she wishes it wishes wish
he desires desires
he wishes wishes
```

If you start a Latin-to-English quiz, the last line of the sample information file instructs the **quiz** command to ask you the meaning of the Latin word `albus`. If you start an English-to-Latin quiz, the **quiz** command displays `white` and accepts `albus`, `alba`, or `album` for the answer.

If any of the characters { (left brace), } (right brace),[(left bracket) ,], (right bracket) or | (vertical bar) appear in a question item, the **quiz** command gives the first alternative of every | group and displays every optional group. Thus, the English-to-Latin question for the fourth definition in this sample is `she desires`.

Files

/usr/games/lib/quiz/index	Default index file for quiz categories.
/usr/games/lib/quiz/*	Used to specify the contents of a given file.
/usr/games	Location of the system's games.

Related Information

The **arithmetic** command, **back** command, **bj** command, **craps** command, **fish** command, **fortune** command, **hangman** command, **moo** command, **number** command, **ttt** command, **turnoff** command, **turnon** command, **wump** command.

---

quot Command

Purpose

Summarizes file system ownership.

Syntax

quot [**-c**] [**-f**] [**-h**] [**-n**][**-v**] [*FileSystem ...*]

quot -a [**-c**] [**-f**] [**-h**] [**-n**] [**-v**]

Description

The **quot** command summarizes file system ownership by displaying the number of 512-byte blocks currently owned by each user in the specified file system (*FileSystem*). If no file system is specified, the **quot** command displays the same information for each of the file systems in the **/etc/filesystems** file.

Flags

-a Generate a report on all mounted systems.

-c Displays a three-column report. The first column specifies the file size in 512-byte blocks. The second column specifies the number of files of that size. Finally, the third column specifies the cumulative total of 512-byte blocks in all files of that size or smaller.
Note: Files greater than or equal to 500 blocks are grouped under a block size of 499. However, their exact block count contributes to the cumulative total of blocks.

-f Displays the total number of blocks, the total number of files, and the user name associated with these totals.

-h Estimates the number of blocks used by the file. This estimation is based on the file size and may return greater than actual block usage when used on files with holes.

-n Produces a list of all files and their owners by running the following pipeline:

```
ncheck filesystem | sort +On | quot -n filesystem
```

-v Displays output in three columns containing the number of blocks not accessed in the last 30, 60, and 90 days.

Security

Access Control: This command is owned by the bin user and bin group.

Examples

1. To display the number of files and bytes owned by each user in the **/usr** file system, enter:

   ```
   quot -f /usr
   ```

   The system displays the following information:

   ```
   /usr:
   63056    3217    bin
   20832     390    root
    1184      42    uucp
      56       5    adm
       8       1    guest
       8       1    sys
   ```

2. To display file size statistics, the number of files of each size, and a cumulative total, enter:

   ```
   quot -c /usr
   ```

   The system displays the following information:

   ```
   /usr:
   8       103      824
   16      2        856
   499     0        856
   ```

3. To generate a report of all mounted file systems, type:

   ```
   quot -a
   ```

4. To generate a report of the **/var** file system, type:

```
#quot -v /var
/var:
45695   root            12852   11878   11774
 2569   guest            2567    1280     960
 2121   adm                92      91      91
 1343   bin               465     233     193
  943   imnadm            937     633     633
   14   uucp                0       0       0
    5   daemon              0       0       0
    1   invscout            1       1       1
    1   nuucp               1       1       1
    1   sys                 0       0       0
```

Files

/etc/passwd	Contains user names.
/etc/filesystems	Contains file system names and locations.

Related Information

The **du** command, **ls** command.

---

quota Command

Purpose

Displays disk usage and quotas.

Syntax

quota [**-u** [*User*]] [**-g** [*Group*]] [**-v** | **-q**]

Description

The **quota** command displays disk usage and quotas. By default, or with the **-u** flag, only user quotas are displayed. The **quota** command reports the quotas of all file systems listed in the **/etc/filesystems** file. If the **quota** command exits with a non-zero status, one or more file systems are over quota.

A root user can use the **-u** flag with the optional *User* parameter to view the limits of other users. Users without root user authority can view the limits of groups of which they are members by using the **-g** flag with the optional *Group* parameter.

Note: If a particular user has no files in a file system on which that user has a quota, this command displays `quota: none` for that user. The user's actual quota is displayed when the user has files in the file system.

Flags

-g Displays the quotas of the user's group.
-u Displays user quotas. This flag is the default option.
-v Displays quotas on file systems with no allocated storage.
-q Prints a terse message, containing only information about file systems with usage over quota.
Note: The **-q** flag takes precedence over the **-v** flag.

Security

Access Control: This command is owned by the root user and the bin group.

Privilege Control: This program is **setuid** in order to allow non-privileged users to view personal quotas.

Examples

1. To display your quotas as user `keith`, type:

   ```
   quota
   ```

   The system displays the following information:

   ```
   User quotas for user keith (uid 502):
   Filesystem  blocks  quota  limit  grace  Files  quota limit grace
           /u      20     55     60            20     60    65
   ```

2. To display quotas as the root user for user `davec`, type:

   ```
   quota -u davec
   ```

   The system displays the following information:

   ```
   User quotas for user davec (uid 2702):
   Filesystem  blocks  quota  limit  grace  files  quota limit grace
           /u      48     50     60             7     60    60
   ```

Files

quota.user	Specifies user quotas.
quota.group	Specifies group quotas.
/etc/filesystems	Contains file system names and locations.

Related Information

Setting Up the Disk Quota System in *AIX 5L Version 5.2 Security Guide*.

The **edquota** command, **quotacheck** command, **quotaon** and **quotaoff** commands, **repquota** command.

---

quotacheck Command

Purpose

Checks file system quota consistency.

Syntax

quotacheck [**-g**] [**-u**] [**-v**] { **-a** | *Filesystem ...* }

Description

The **quotacheck** command examines a file system specified by the *FileSystem* parameter, builds a table of current disk usage, and compares the information in the table to that recorded in the file system's disk quota file. If any inconsistencies are detected, the quota files are updated. If an active file system is checked, the current system copy of the incorrect quotas is updated as well. By default, both user and group quotas are checked.

The optional **-g** flag specifies that only group quotas are checked. The optional **-u** flag specifies that only user quotas are checked. Specifying both **-g** and **-u** flags is equivalent to the default option. The **-a** flag specifies that all file systems with disk quotas in the **/etc/filesystems** file are checked.

The **quotacheck** command usually operates silently. If the **-v** flag is specified, the **quotacheck** command reports discrepancies between the calculated and recorded disk quotas.

The **quotacheck** command expects each file system to contain the **quota.user** and **quota.group** files, located at the root of the associated file system. These default file names can be changed in the **/etc/filesystems** file. If these files do not exist, the **quotacheck** command creates them.

Note: Do not run the quotacheck command against an active file system. If the file system has any current activity, running **quotacheck** may result in incorrect disk usage information.

Flags

-a Checks all file systems with disk quotas and read-write permissions indicated by the **/etc/filesystems** file.
-g Checks group quotas only.
-u Checks user quotas only.
-v Reports discrepancies between the calculated and recorded disk quotas.

Security

Access Control: Only a user with root user authority can execute this command.

Examples

1. To check the user and group quotas in the `/usr` file system, type:
   ```
   quotacheck /usr
   ```
2. To check only the group quotas in the `/usr` file system, type:
   ```
   quotacheck -g /usr
   ```

Files

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

Related Information

The **edquota** command, **quota** command, **quotaon** and **quotaoff** commands, **repquota** command, .

Setting Up the Disk Quota System in *AIX 5L Version 5.2 Security Guide*.

---

quotaon or quotaoff Command

Purpose

Turns on and off file system quotas.

Syntax

quotaon [**-g**] [**-u**] [**-v**] { **-a** | *FileSystem ...* }

quotaoff [**-g**] [**-u**] [**-v**] { **-a** | *FileSystem ...* }

Description

The **quotaon** command enables disk quotas for one or more file systems specified by the *FileSystem* parameter. The specified file system must be defined with quotas in the **/etc/filesystems** file, and must be mounted. The **quotaon** command looks for the **quota.user** and **quota.group** default quota files in the root directory of the associated file system. These file names may be changed in the **/etc/filesystems** file.

By default, both user and group quotas are enabled. The **-u** flag enables only user quotas; the **-g** flag enables only group quotas. Specifying both **-g** and **-u** flags is equivalent to the default option. The **-a** flag specifies that all file systems with disk quotas, as indicated by the **/etc/filesystems** file, are enabled.

The **quotaoff** command disables disk quotas for one or more file systems. By default, both user and group quotas are disabled. The **-a**, **-g**, and **-u** flags operate as with the **quotaon** command.

The **-v** flag prints a message for each file system in which quotas are turned on or off with the **quotaon** and **quotaoff** commands, respectively.

Flags

-a Enables or disables all file systems that are read-write and have disk quotas, as indicated by the **/etc/filesystems** file. When used with the **-g** flag, only group quotas in the **/etc/filesystems** file are enabled or disabled; when used with the **-u** flag, only user quotas in the **/etc/filesystems** file are enabled or disabled.
-g Specifies that only group quotas are enabled or disabled.
-u Specifies that only user quotas are enabled or disabled.
-v Prints a message for each file system in which quotas are turned on or off.

Security

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

Examples

1. To enable user quotas for the `/usr` file system, enter:
   ```
   quotaon -u /usr
   ```
2. To disable user and group quotas for all file systems in the **/etc/filesystems** file and print a message, enter:
   ```
   quotaoff -v -a
   ```

Files

quota.user	Specifies user quotas.
quota.group	Specifies group quotas.
/etc/filesystems	Contains file system names and locations.

Related Information

The **quota** command, **edquota** command, **repquota** command, **quotacheck** command.

The Disk Quota System Overview in *AIX 5L Version 5.2 Security Guide* introduces the disk quota system.

Setting Up the Disk Quota System in *AIX 5L Version 5.2 Security Guide*.

ranlib Command

Purpose

Converts archive libraries to random libraries.

Syntax

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

Description

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

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

Flags

-t Touches the named archives without modifying them.

-X *mode* Specifies the type of object file **ranlib** 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

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 **ranlib** to process any 64-bit objects and ignore 32-bit objects. The **-X** flag overrides the **OBJECT_MODE** variable.

Examples

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

```
ranlib genlib.a
```

Files

/usr/ccs/bin/ranlib Contains the **ranlib** command.

Related Information

The Subroutines Overview in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

The **ld** command, **ar** command, **lorder** command, **make** command.

ras_logger Command

Purpose

Log an error using the errors template.

Syntax

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

Description

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

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

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

Flags

-y *template-file* — Specifies a template file other than the **/var/adm/ras/errtmplt** default file.

Examples

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

```
+ FOO:
 Catname = "foo.cat"
 Err_Type = TEMP
 Class = O
 Report = TRUE
 Log    = TRUE
 Alert  = FALSE
 Err_Desc = {1, 1, "Error FOO"}
 Prob_Causes = {1, 2, "Just a test"}
 User_Causes = {1, 2, "Just a test"}
 User_Actions = {1, 3, "Do nothing"}
 Detail_Data = 4, {2, 1, "decimal"} ,DEC
 Detail_Data = W, {2, 1, "hex data"} ,HEX
 Detail_Data = 100, {2, 1, "long string"} ,ALPHA
```

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

```
FOO
resource
0
15
A0
hello world
```

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

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

```
+ FOO:
  Catname = "foo.cat"
  Err_Type = TEMP
  Class = O
  Report = TRUE
```

```
Log    = TRUE
Alert  = FALSE
Err_Desc = {1, 1, "Error FOO"}
Prob_Causes = {1, 2, "Just a test"}
User_Causes = {1, 2, "Just a test"}
User_Actions = {1, 3, "Do nothing"}
Detail_Data = 8, {2, 1, "decimal"} ,DEC
Detail_Data = W, {2, 1, "hex data"} ,HEX
Detail_Data = 100, {2, 1, "long string"} ,ALPHA
```

The **ras_logger** command inputs the following into the **tfile**file:

```
FOO
resource
0
15 -15
A0
hello world
```

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

Related Information

The **errpt** and **errlogger** commands.

---

rc Command

Purpose

Performs normal startup initialization.

Syntax

rc

Description

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

Notes:

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

Related Information

The **fsck** command, **init** command, **mount** command.

---

rc.mobip6 Command

Purpose

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

Syntax

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

Description

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

Flags

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

Exit Status

0 The command completed successfully.

>0 An error occurred.

Security

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

Examples

1. The following example enables the system as a mobile IPv6 home agent and correspondent node:
   ```
   /etc/rc.mobip6 start -H
   ```
2. The following example enables the system as a mobile IPv6 correspondent node and enables IP security checking:
   ```
   /etc/rc.mobip6 start -S
   ```
3. The following example disables all mobile IPv6 and IPv6 gateway functionality on the system:
   ```
   /etc/rc.mobip6 stop -N -F
   ```
4. The following example disables all mobile IPv6 functionality but allows the system to continue functioning as an IPv6 gateway:
   ```
   /etc/rc.mobip6 stop
   ```

Files

/etc/rc.mobip6	Contains the **rc.mobip6** command.

Related Information

The **kmodctrl** command, **mobip6reqd** command, **mobip6ctrl** command, **ndpd-router** command.

The Mobile IPv6 section in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

rc.powerfail Command

Purpose

Shuts down a system when a power failure is detected.

Syntax

rc.powerfail [**-h**] | [[**-s**] [**-t** [*mm*]]]

Description

Power related events are handled by the **rc.powerfail** script to achieve graceful shutdown of the operating system whenever possible. Note that if a critical power failure condition occurs and if battery backup is present, the operating system can take certain actions before the battery drains.

The **rc.powerfail** command is started by the **/etc/inittab** file when **init** receives a **SIGPWR** signal from the kernel. The **rc.powerfail** command first calls the **machstat** command to determine the state of the system. The **rc.powerfail** command should be called only when a power failure or fan fault has occurred.

In the case of a severe failure without any form of backup power supply, the **rc.powerfail** command is given approximately 20 seconds prior to halting the system. The severe failure situations include loss of primary power, thermal high, internal failure, or power supply overload. In these cases, **rc.powerfail** warns the users currently logged on to the system and then immediately halts the system.

In case of a critical failure, the **rc.powerfail** command waits approximately 10 minutes before starting the shut down of the system. The user can alter this time by using the **-t** option on the **/etc/inittab** file powerfail entry. Prior to the last 60 seconds, logged-on users are sent messages telling them how much time remains until shutdown. If a shutdown is NOT desired, the user may add the **-s** option to the command in the **/etc/inittab** file.

Flags

-h Gives an information message containing the power status codes and the resulting action.

-s Does not do a system shutdown if there is a power failure in systems with either a battery backup or fan fault. The logged-on users still receive all the appropriate messages, but the actual system shutdown is left up to the system administrator. This flag has no effect if a critical power failure is detected.

-t*mm* Gives the number of whole minutes until system shutdown in the case of a primary power loss with battery backup or fan fault. This number should be equal to half the length of time guaranteed by the battery backup. This flag has no effect if a critical power failure is detected.

Exit Status

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

0 Normal condition.

1 Syntax error.

2 **halt -q** failed

3 **shutdown -F** failed.

4 An error has occurred. Shut your system down immediately using **shutdown -F**.

5 An undefined state. Call your Service Representative.

Security

Access Control: root only.

Examples

1. The various messages related to power status codes can be displayed by invoking **rc.powerfail** as follows:

   ```
   rc.powerfail -h
   ```

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

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

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

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

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

   The next critical **SIGPWR** received by **init** will have a 2 minute delay until system shutdown.

Files

/etc/inittab	Controls the initialization process.

Related Information

The **machstat** command, **init** command, **chitab** command, **halt** command, **shutdown** command.

---

rcp Command

Purpose

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

Syntax

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

Description

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

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

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

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

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

File Permissions and Ownership

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

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

Using Standard Authentication

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

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

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

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

For Kerberos 5 Authentication

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

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

rcp and Named Pipelines

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

Restrictions

The SP Kerberos V4 rcp execution path does not support remote-to-remote copy as Kerberos does not support forwarding credentials. The message you would receive under these circumstances is the message indicating you do not have tickets and must use **kinit** to login. The message would be issued from the remote source machine. Please see the example below for using Kerberos to perform a remote-to-remote copy.

Flags

-p Preserves the modification times and modes of the source files in the copies sent to the destination only if the user has root authority or is the owner of the destination. Without this flag, the **umask** command at the destination modifies the mode of the destination file, and the modification time of the destination file is set to the time the file is received.

When this flag is not used, the umask being honored is the value stored in the appropriate database. It is not the value that is set by issuing the **umask** command. The permission and ownership values that result from the **umask** command do not affect those stored in the database.

-r Recursively copies, for directories only, each file and subdirectory in the source directory into the destination directory.

-F Causes the credentials to be forwarded. In addition, the credentials on the remote system will be marked forwardable (allowing them to be passed to another remote system). This flag will be ignored if Kerberos 5 is not the current authentication method. Authentication will fail if the current DCE credentials are not marked forwardable.

-k *realm* Allows the user to specify the realm of the remote station if it is different from the local systems realm. For these purposes, a realm is synonymous with a DCE cell. This flag will be ignored if Kerberos 5 is not the current authentication method.

Security

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

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

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

Parameters

Host:*File* Specifies the host name (*Host*) and file name (*File*) of the remote destination file, separated by a : (colon).
Note: Because the **rcp** command assumes that a : (colon) terminates a host name, you must insert a \ (backslash) before any colons that are embedded in the local file and directory names.

User@*Host*:*File* Specifies the user name (*User*@) that the **rcp** command uses to set ownership of the transferred file, the host name (*Host*), and file name (*File*) of the remote destination file. The user name entered for the remote host determines the file access privileges the **rcp** command uses at that host.

File Specifies the file name of the local destination file.

Host:*Directory* Specifies the host name (*Host*) and directory name (*Directory*) of the remote destination directory.
Note: Because the **rcp** command assumes that a : (colon) terminates a host name, you must insert a \ (backslash) before any colons that are embedded in the local file and directory names.

User@*Host*:*Directory* Specifies the user name (*User*@) the **rcp** command uses to set ownership of the transferred file, the host name (*Host*), and directory name (*Directory*) of the remote destination directory. The user name entered for the remote host determines the file access privileges the **rcp** command uses at that host.

Directory The directory name of the local destination directory.

Exit Status

This command returns the following exit values:

0 Successful completion.

>0 An error occurred.

Examples

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

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

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

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

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

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

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

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

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

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

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

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

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

Files

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

Prerequisite Information

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

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

Related Information

The **cp** command, **ftp** command, **rlogin** command, **rsh** command, **telnet** command, **tftp** command, **umask** command.

SP Commands: **k4init**, **k4list**, **k4destroy**, **lsauthpar**, **chauthpar**, **Kerberos**

Environment variable: **KRBTKFILE**

The **rshd** and **krshd** daemon.

The **kvalid_user** function.

The **.rhosts** file format, **hosts.equiv** file format.

Network Overview in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

Secure Rcmds in *AIX 5L Version 5.2 System User's Guide: Communications and Networks*.

---

rcvdist Command

Purpose

Sends a copy of incoming messages to additional recipients.

Syntax

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

Description

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

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

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

Flags

-form *File*	Specifies the file that formats the command output. The default is the **rcvdistcomps** file.
-help	Lists the command syntax, available switches (toggles), and version information.

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

Files

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

Related Information

The **ali** command, **rcvpack** command, **rcvstore** command, **rcvtty** command, **sendmail** command, **slocal** command, **whom** command.

The **.mh_alias** file format.

Mail Overview in *AIX 5L Version 5.2 System User's Guide: Communications and Networks*.

rcvpack Command

Purpose
Saves incoming messages in a packed file.

Syntax
rcvpack [*File*]

Description
The **rcvpack** command places incoming messages in the packed file specified by the *File* parameter. The **rcvpack** command is not started by the user. The **rcvpack** command is placed in the **$HOME/.maildelivery** file called by the **/usr/lib/mh/slocal** command.

The **rcvpack** command appends a copy of the incoming message to the specified file and runs the **packf** command on the file. Specifying the **rcvpack** command in the **.maildelivery** file runs the **rcvpack** command on all incoming messages.

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

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

Files

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

Related Information
The **inc** command, **packf** command, **rcvdist** command, **rcvstore** command, **rcvtty** command, **sendmail** command, **slocal** command.

The **.mh_alias** file format.

Mail Overview in *AIX 5L Version 5.2 System User's Guide: Communications and Networks*.

rcvstore Command

Purpose
Incorporates new mail from standard input into a folder.

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

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

The **rcvstore** command accepts messages from standard input and places them in a specified folder. Run the **rcvstore** command on all incoming messages by specifying the **rcvstore** command in the **.maildelivery** file.

You can specify **rcvstore** command flags in the **$HOME/.maildelivery** file or, as with most MH commands, in the **$HOME/.mh_profile** file.

Flags

-create	Creates the specified folder in your mail directory if the folder does not exist. This flag is the default.
+*Folder*	Places the incorporated messages in the specified folder. The default is +inbox.
-help	Lists the command syntax, available switches (toggles), and version information. **Note:** For MH, the name of this flag must be fully spelled out.
-nocreate	Does not create the specified folder if the folder does not exist.
-nopublic	Restricts the specified sequence of messages to your usage. The **-nopublic** flag does not restrict the messages in the sequence, only the sequence. This flag is the default if the folder is write-protected against other users.
-nozero	Appends the messages incorporated by the **rcvstore** command to the specified sequence of messages. This flag is the default.
-public	Makes the specified sequence of messages available to other users. The **-public** flag does not make protected messages available, only the sequence. This flag is the default if the folder is not write-protected against other users.
-sequence *Name*	Adds the incorporated messages to the sequence of messages specified by the *Name* parameter.
-zero	Clears the specified sequence of messages before placing the incorporated messages into the sequence. This flag is the default.

Profile Entries

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

Files

$HOME/.mh_profile	Contains the MH user profile.
$HOME/.maildelivery	Provides the user with MH instructions for local mail delivery.
$HOME/.forward	Provides the user with the default message filter.

Related Information

The **inc** command, **rcvdist** command, **rcvpack** command, **rcvtty** command, **sendmail** command, **slocal** command.

The **.mh_alias** File Format.

Mail Overview in *AIX 5L Version 5.2 System User's Guide: Communications and Networks*.

rcvtty Command

Purpose
Notifies the user of incoming messages.

Syntax
rcvtty [*Command*]

Description
The **rcvtty** command sends the user a message that incoming mail has arrived. The **rcvtty** command is not started by the user. The **rcvtty** command is placed in the **.maildelivery** file called by the **/usr/lib/mh/slocal** file.

The **rcvtty** command sends a one-line scan listing to your terminal. If you specify a command in the *Command* parameter, the **rcvtty** command executes that command using the incoming message as the standard input and sending the output to the terminal. For the **rcvtty** command to write output to your terminal, the write permissions for your terminal must be set to `all`.

Run the **rcvtty** command on all incoming messages by specifying `rcvtty` in the **.maildelivery** file.

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

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

Files
$HOME/.maildelivery	Provides the user with instructions for local mail delivery.
$HOME/.forward	Provides the user with the default message filter.
$HOME/.mh_profile	Contains the MH user profile.

Related Information
The **rcvdist** command, **rcvpack** command, **rcvstore** command, **sendmail** command, **slocal** command.

The **.mh_alias** file format.

Mail Overview in *AIX 5L Version 5.2 System User's Guide: Communications and Networks*.

rdist Command

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

Purpose
Remote file distribution client program

Syntax

To Use a Distribution File

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

To Interpret Arguments as a Small Distribution File

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

Description

Attention: Do not attempt to use the **rdist** command to send a file larger than 2 Gigabytes in size to a pre-AIX 4.2 or non-AIX machine. Doing so will result in undefined behaviors and, in rare cases, the loss of data.

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

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

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

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

The **rdist** command requires that a **.rhosts** file be configured on each host. See ″.rhosts File Format for TCP/IP″ in *AIX 5L Version 5.2 Files Reference* for details.

Notes:

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

Flags

-b Performs a binary comparison and updates files if they differ.

-c Directs the **rdist** command to interpret the remaining arguments as a small distribution file. Available arguments are:

Name Specifies single name or list of names separated by blanks. The value can be either a file or a subcommand.

[*Login***@]***Host*
Specifies the machine to be updated and, optionally, the login name to be notified of the update.

Destination
Specifies a file on the remote machine if a single name is specified in the *Name* argument; specifies a directory if more than one name is specified.

Note: Do not use the **-c** flag with the **-f**, **-d**, or **-m** flag.

-d *Argument***=***Value* Defines the *Argument* variable as having the value specified by the *Value* variable. The **-d** flag defines or overrides variable definitions in the **distfile** file. The *Value* variable can be specified as an empty string, one name, or a list of names surrounded by parentheses and separated by tabs or spaces.

-D Turns on the debugging output.

-f *FileName* Specifies the name of the distribution file. If you do not use the **-f** flag, the default value is the **distfile** or **Distfile** file in your **$HOME** directory.

-h Copies the file that the link points to rather than the link itself.

-i Ignores unresolved links. The **rdist** command maintains the link structure of files being transferred and warns users if it cannot find all the links.

-m *Host* Limits which machines are to be updated. You can use the **-m** *Host* option multiple times to limit updates to a subset of the hosts listed in the **distfile** file.

-n Prints the subcommands without executing them. Use the **-n** flag to debug the **distfile** file.

-q Operates in quiet mode. The **-q** option suppresses printing of modified files on standard output.

-R Removes extraneous files. If a directory is being updated, any files that exist on the remote host but not in the master directory are removed. Use the **-R** flag to maintain identical copies of directories.

-v Verifies that the files are up-to-date on all hosts; files that are out-of-date are then displayed. However, the **rdist -v** command neither changes files nor sends mail. This flag overrides the **-b** flag when they are used together.

-w Appends the entire path name of the file to the destination directory name. Normally, the **rdist** command uses only the last component of a name for renaming files, preserving the directory structure of the copied files.

When the **-w** flag is used with a file name that begins with a ~ (tilde), everything except the home directory is appended to the destination name. File names that do not begin with a / (slash) or a ~ (tilde) use the destination user's home directory as the root directory for the rest of the file name.

-y Prevents recent copies of files from being replaced by files that are not as recent. Files are normally updated when their time stamp and size differ. The **-y** flag prevents the **rdist** command from updating files more recent than the master file.

Distribution File (distfile File)

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

Entry Formats

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

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

[*Label***:**] *SourceList* **->** *DestinationList SubcommandList*

	Directs the **rdist** command to distribute files named in the *SourceList* variable to hosts named in the *DestinationList* variable. Distribution file commands perform additional functions.
[*Label*:] *SourceList* **::** *TimeStampFile SubcommandList*	Directs the **rdist** command to update files that have changed since a given date. Distribution file subcommands perform additional functions. Each file specified with the *SourceList* variable is updated if the file is newer than the time-stamp file. This format is useful for restoring files.

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

Entries

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

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

Source and Destination List Format

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

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

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

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

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

Distribution File Subcommands

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

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

install *Options* [*OptionalDestName*]**;**
Copies out-of-date files and directories. The **rdist** command copies each source file or directory to each host in the destination list. The available options as specified by the *Options* variable are the **rdist** command flags **-b**, **-h**, **-i**, **-R**, **-v**, **-w**, and **-y**. These options only apply to the files specified by the *SourceList* variable. When you use the **-R** flag, nonempty directories are removed if the corresponding file name is absent on the master host. The *OptionalDestName* parameter renames files.

If no **install** subcommand appears in the subcommand list or the destination name is not specified, the source file name is used. Directories in the path name are created if they do not exist on the remote host. The login name used on the destination host is the same as the local host unless the destination name is of the format *login@host*.

notify *NameList***;**
Mails the list of updated files and any errors that may have occurred to the listed names (the *NameList* parameter). If no **@** (at sign) appears in the name, the destination host is appended to the name (*name@host*).

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

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

special *NameList "String"***;**
Specifies shell commands (the *"String"* variable) to be executed on the remote host after the file specified by the *NameList* variable is updated or installed. If the *NameList* variable is omitted, the shell commands are executed for every file updated or installed. The shell variable **FILE** is set to the current file name before the **rdist** command executes the *"String"* variable. The *"String"* value must be enclosed in " " (double quotation marks) and can cross multiple lines in the distribution file.

Exit Status

This command returns the following exit values:

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

Examples

Examples of the Format: VariableName = NameList

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

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

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

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

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

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

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

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

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

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

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

   or

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

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

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

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

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

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

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

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

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

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

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

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

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

Files

/usr/bin/rdist	Contains the **rdist** command.
$HOME/distfile	Contains a list of subcommands to be read by the **rdist** command.
/tmp/rdist	Contains an update list. This is a temporary file.

Related Information

The **csh** command, **ksh** command, **sh** command.

The **.rhosts** file format.

Network Overview in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

usr/sbin/rdist Command

Purpose

Client program for distributing files remotely.

Syntax

To Use a Distribution File

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

To Interpret Arguments as a Small Distribution File

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

To Invoke the Old rdist as a Server

/usr/sbin/rdist -Server

For Version Information

/usr/sbin/rdist -V

Description

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

- The distribution file **distfile** in the current directory.
- The standard input if **distfile** is specified as -.
- If the **-f** flag is not used, **rdist** looks for the file named *distfile* and *Distfile*.
- If the **-c** flag is used, the trailing arguments are interpreted as a small **distfile**. The equivalent **distfile** is as follows.

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

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

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

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

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

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

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

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

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

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

```
rdistd -S
```

or

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

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

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

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

Flags

-A *num*	Update or install files only if a minimum number of free files (inodes) exists on a filesystem.
-a *num*	Update or install files only if a minimum amount of free space exists on a filesystem.
-d *var = value*	Assign *value* to variable *var*. This option is used to define or override variable definitions in the **distfile**. *Value* can be the empty string, one name, or a list of names surrounded by parentheses and separated by tabs and/or spaces.
-F	Update all clients sequentially without forking child processes.
-f *distfile*	Use **distfile** as the distribution file. If **distfile** is specified as -, read from standard input.
-l *logopts*	Sets local logging options. See the **Message Logging** section for more information on the syntax for *logopts*.
-L *logopts*	Sets remote logging options. *logopts* is the same as for local logging except the values are passed to the remote server (**rdistd**). See the **Message Logging** section for more information on the syntax of *logopts*.
-M *num*	Limit the maximum number of simultaneously running child **rdist** processes to *num*. The default is 4.
-m *machine*	Limits the updating of files to the given machine. Multiple **-m** arguments can be given to limit updates to a subset of the hosts listed in the **distfile**.
-n	Display but do not execute commands. Use the **-n** flag to debug **distfile**.

-o *distopts*

Specifies the dist options to enable. *distopts* is a comma separated list of options listed below. The valid values for *distopts* are:

chknfs If the target filesystem is NFS, do not check or update files.

chkreadonly
If a file on the target host resides on a read only filesystem, no checking or updating of the file is attempted.

chksym
If the target on the remote host is a symbolic link, but is not on the master host, the remote target will be left a symbolic link.

compare
Perform a binary comparison and update files if they differ.

follow Copy the file that the symbolic link points to rather than the link itself.

ignlnks
Ignore links which do not resolve. The normal behavior of **rdist** is to warn the user about unresolved links.

nochkowner
If the file already exists, do not check user ownership. The file ownership is only set when the file is updated.

nochkgroup
If the file already exists, do not check group ownership. The file ownership is only set when the file is updated.

nochkmode
Avoid checking file and directory permission modes. The permission mode is only set when the file is updated.

nodescend
Do not descend recursively into a directory. Only the existence, ownership, and mode of the directory are checked.

noexec
Do not check or update executable files that are in **a.out** format.

numchkgroup
Use the numeric group id (gid) to check group ownership instead of the group name.

numchkowner
Use the numeric user id (uid) to check user ownership instead of the user name.

quiet Supress printing files that are being modified on the standard output.

remove
Remove any files in directories that exist on the remote host that do not exist in the master directory on the local host.

savetargets
Save files that are updated instead of removing them. Target files that are updated are first renamed from **filename** to **filename.OLD**.

sparse Enable checking for sparse files. This option adds some additional processing overhead so it should only be enabled for targets likely to contain sparse files.

-o *distopts* *(dist options, continued)*:

verify Any file on any host that is out of date will be displayed but no file will be changed nor any mail sent.

whole The whole file name is appended to the destination directory name. Normally, only the last component of a name is used when renaming files. This will preserve the directory structure of the files being copied instead of flattening the directory structure. For example, rdisting a list of files such as **/path/dir1/f1** and **/path/dir2/f2** to **/tmp/dir** would create files **/tmp/dir/path/dir1/f1** and **/tmp/dir/path/dir2/f2** instead of **/tmp/dir/dir1/f1** and **/tmp/dir/dir2/f2**.

younger
Files are normally updated if their *mtime* and *size* disagree. This option causes **rdist** not to update files that are younger than the master copy. This can be used to prevent newer copies on other hosts from being replaced. A warning message is printed for files which are newer than the master copy.

-p <*rdist-path*> Search for the **rdistd** server in the given path on the target host.

-P <*rdist-path*> Use the transport program as given in *transport-path*. The *transport-path* may be a colon seperated list of possible pathnames. In this case, the first component of the path to exist is used.

-t *timeout* Sets the *timeout* period (in seconds) for waiting for responses from the remote **rdist** server. The default is 900 seconds.

-V Prints the version information and exits.

Message Logging

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

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

The form of *logopts* should be the following:

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

The valid facility names are as follows:

stdout
Messages to standard output.

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

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

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

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

change
Log messages for things that change.

info Log general information.

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

nerror Log messages for normal errors that are not fatal.

ferror Log messages for fatal errors.

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

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

debug Log debugging information.

all Log all but debug messages.

The Distribution File

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

Entry Formats

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

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

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

Entries

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

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

Source and Destination List Format

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

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

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

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

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

Distribution File Subcommands

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

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

install Options[OptionalDestName];	Copies out-of-date files and directories. The **rdist** command copies each source file or directory to each host in the destination list. The available options as specified by the *Options* variable are the **rdist** command flags **-b**, **-h**, **-i**, **-R**, **-v**, **-w**, and **-y**. These options only apply to the files specified by the *SourceList* variable. When you use the **-R** flag, nonempty directories are removed if the corresponding file name is absent on the master host. The *OptionalDestName* parameter renames files. If no install subcommand appears in the subcommand list or the destination name is not specified, the source file name is used. Directories in the path name are created if they do not exist on the remote host. The login name used on the destination host is the same as the local host unless the destination name is of the format login@host.
notify NameList;	Mails the list of updated files and any errors that may have occurred to the listed names (the *NameList* parameter). If no @ (at sign) appears in the name, the destination host is appended to the name (name@host).
except NameList;	Causes the **rdist** command to update all the files specified by the *SourceList* entry except for those files specified by the *NameList* variable.
except_pat NameList;	Prevents the **rdist** command from updating any files that contain a string that matches a member of the list specified by the *NameList* variable.

special NameList ″String″;	Specifies shell commands (the ″String″ variable) to be executed on the remote host after the file specified by the *NameList* variable is updated or installed. If the *NameList* variable is omitted, the shell commands are executed for every file updated or installed. The shell variable FILE is set to the current file name before the **rdist** command executes the ″String″ variable, the variable REMFILE will contain the full pathname of the remote file that was just updated and the variable BASEFILE will contain the basename of the remote file that was just updated. The ″String″ value must be enclosed in double quotation marks and can cross multiple lines in the distribution file.
cmdspecial NameList ″String″;	The **cmdspecial** command is similar to the **special** command, except it is executed only when the entire command is completed instead of after each file is updated. The shell variable FILES will contain the list of files. Each file name in the FILES shell variable is separated by a colon.

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

Exit Status

This command returns the following exit values:

0 Successful completion.
>0 An error occurred.

Examples

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

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

   In the above example, the HOSTS variable is defined to be matisse and root@arpa. The **rdist** command updates files on the hosts matisse and root@arpa. You could use this variable as a destination list.
2. To indicate a name to use as a value for a SourceList entry, enter a line similar to the following:

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

   In the above example, the FILES value is defined to be the files to be used for the *SourceList* entry.
3. To indicate which files to exclude from the updating process, enter a line similar to the following:

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

   In the above example, the EXLIB value is defined as a list of files to exclude from the updating process.
4. To copy all files from /usr/src/bin to arpa expanding the namelist variable so that all files except those present in the namelist variable and having .o as an extension are copied:

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

   or

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

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

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

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

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

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

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

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

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

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

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

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

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

Files

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

Related Information

The **csh** command, **rcmd()** system call, **rdistd** command,**rsh** command, **sh** command, **stat()** system call.

---

rdistd Command

Purpose

Server program for distributing files remotely.

Syntax

rdistd -S

rdistd -V

Description

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

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

Flags

-V Print version information and exit.

Exit Status

This command returns the following exit values:

0 Successful completion.

>0 An error occurred.

Files

/usr/sbin/rdistd	Containsthe **rdistd** server
/usr/bin/rdistd	Symbolic link to **/usr/sbin/rdistd**

Related Information

"rdist Command" on page 481, and "rsh or remsh Command" on page 645.

---

rdump Command

Purpose

Backs up files onto a remote machine's device.

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

Syntax

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

Description

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

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

Notes:

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

Flags

-b *Blocks*	Specifies the number of blocks to write in a single output operation. If you do not specify the *Blocks* variable, the **rdump** command uses a default value appropriate for the physical device selected. Larger values of the *Blocks* variable result in larger physical transfers to tape devices.
-B	Terminates the command without querying the user when an error occurs. If you specify the **-B** flag, the **rdump** command returns a nonzero value.
-c	Specifies that the tape is a cartridge format, not a 9-track format.
-d *Density*	Specifies the density of the tape in bits-per-inch (bpi). This value is used in calculating the amount of tape used per volume. If you do not specify a value for the *Density* variable, the default density is 1600 bpi. When using the **-c** flag without specifying a tape density, the default density is 8000 bpi.
-f *Machine***:***Device*	Specifies the *Machine* variable as the hostname of the remote machine. To send output to the named device, specify the *Device* variable as a file name (such as the **/dev/rmt0** file). The *Device* variable should specify only tape devices.
-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, type the following: `-L 2g`. This flag only applies to AIX 4.2 and above.
-s *Size*	Specifies the size of the tape in feet using the *Size* variable. If you do not specify a tape size, the default size is 2300 feet. When using the **-c** flag without specifying a tape size, the default size is 1700 feet. When the tape drive reaches the specified size, the **rdump** command waits for the tape to be changed.
-u	Updates the time, date, and level of the remote backup in the **/etc/dumpdates** file. This file provides the information needed for maintaining incremental backups.
-w	Currently disabled.
-W	Displays the file systems found in the **/etc/dumpdates** files.
-*Level*	Specifies the remote backup level (0 to 9). The default value of the *Level* variable is 9.
-?	Displays the usage message.

Parameters

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

Exit Status

This command returns the following exit values:

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

Examples

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

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

   The **-u** flag tells the system to update the current backup level records in the **/etc/dumpdates** file. The **-***Level* flag set to backup level 0 specifies that all the files in the **/usr** directory are to be backed up. The ID of the remote machine is `canine` and the device is the **/dev/rmt0** device.
2. To back up files in the **/usr** directory on your local machine to a remote machine using an 8mm, 2.3GB tape, type:

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

   Note: 2.2GB is used here instead of 2.3GB to avoid hitting the actual end of the tape.
3. To back up files in the **/usr** directory on your local machine to a remote machine using 0.25-inch tape, type:

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

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

Files

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

Related Information

The **find** command, **dd** command, **backup** command, **restore** command, **rrestore** command.

The **dumpdates** file, **filesystems** file, **rmt** special file.

Backup Overview for System Management, Directory Overview, Files Overview in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices*.

File Systems in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices*.

---

read Command

Purpose

Reads one line from standard input.

Syntax

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

[*VariableName* ...]

Description

The **read** command reads one line from standard input and assigns the values of each field in the input line to a shell variable using the characters in the IFS (Internal Field Separator) variable as separators. The *VariableName* parameter specifies the name of a shell variable that takes the value of one field from

the line of input. The first shell variable specified by the *VariableName* parameter is assigned the value of the first field, the second shell variable specified by the *VariableName* parameter is assigned the value of the second field, and so on, until the last field is reached. If the line of standard input has more fields than there are corresponding shell variables specified by the *VariableName* parameter, the last shell variable specified is given the value of all the remaining fields. If there are fewer fields than shell variables, the remaining shell variables are set to empty strings.

> **Note:** If you omit the *VariableName* parameter, the variable REPLY is used as the default variable name.

The setting of shell variables by the **read** command affects the current shell execution environment.

Flags

-p Reads input from the output of a process run by the Korn Shell using |& (pipe, ampersand).

> **Note:** An end-of-file character with the **-p** flag causes cleanup for this process to so that another can be spawned.

-r Specifies that the read command treat a \ (backslash) character as part of the input line, not as a control character.

-s Saves the input as a command in the Korn Shell history file.

-u [*n*] Reads input from the one-digit file descriptor number, *n*. The file descriptor can be opened with the ksh exec built-in command. The default value of the *n* is 0, which refers to the keyboard. A value of 2 refers to standard error.

Parameters

VariableName?Prompt specifies the name of one variable, and a prompt to be used. When the Korn Shell is interactive, it will write the prompt to standard error, and then perform the input. If *Prompt* contains more than one word, you must enclose it in single or double quotes.

VariableName... specfies one or more variable names separated by white space.

Exit Status

This command returns the following exit values:

0 Successful completion.

>0 Detected end-of-file character or an error occurred.

Examples

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

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

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

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

   The system displays:

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

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

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

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

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

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

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

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

Related Information

The **printf** command.

The **ksh** command.

---

readlvcopy Command

Purpose

Reads a specific mirror copy of a logical volume.

Syntax

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

Description

Flags

-d *device*	logical volume special device file to be read from
-c *copy*	Requested mirror copy to read from. Vaild values are 1, 2, or 3 for the first, second, or third copy of the data. Data is read even if the logical partition has been marked stale. The default is the first copy of the data.
-C *copy*	Requested mirror copy to read from. Valid values are 1, 2, or 3 for the first, second, or third copy of the data. Stale logical partitions are not read.
-b	Read mirror copy marked as online backup.
-n *number_of_blocks*	Number of 128K blocks to read
-o *outfile*	Destination file. The default is *stdout*
-s *skip*	Number of 128K blocks to skip into *device*.
-S *seek*	Number of 128K blocks to seek into *outfile*

Related Information

The **chlvcopy** command.

---

reboot or fastboot Command

Purpose

Restarts the system.

Syntax

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

Description

The **reboot** command can be used to perform a reboot operation if no other users are logged into the system. The **shutdown** command is used when the system is running and multiple users are logged into the system. Both commands write an entry to the error log that states that the system was deliberately temporarily disabled.

The **reboot** command causes the disks to be synchronized with the **sync** command and allows the system to perform other shutdown activities such as resynchronizing the hardware time-of-day clock. A reboot operation is then started. By default, the system starts and the disks are automatically checked. If successful, the system comes up in the default runlevel.

To force the reboot, press keys that are specifically reserved for rebooting, according to your machine type. If all else fails, turn the machine off, wait at least 60 seconds for the disks to stop spinning, and turn it back on.

The system restarts itself when power is turned on or after the system crashes if the autorestart attribute value is **true**. To find the value of the autorestart attribute, use the **lsattr** command and enter `lsattr -D -l sys0`. The default value is **true**. To reset the autorestart attribute value to **false**, use the **chdev** command and enter:

```
chdev -l sys0 -a autorestart=false.
```

After performing diagnostic tests, the system ROM looks for a bootstrap record. The system loops until it finds one. Normally, the bootstrap record reads the boot program from diskette or fixed disk. If the debugger was configured into the kernel, it is entered first, but an implicit **go** is performed if nothing is entered within about 30 seconds. A carriage return also performs the **go**.

The **reboot** command normally logs the reboot operation using the **syslog** command and places a shutdown record in **/var/adm/wtmp**, the login accounting file. These actions are inhibited if the **-l**, **-n**, or **-q** flags are present.

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

Flags

-l Does not log the reboot or place a shutdown record in the accounting file. The **-l** flag does not suppress accounting file update. The **-n** and **-q** flags imply **-l**.

-n Does not perform the **sync** command. Use of this flag can cause file system damage.

-q Restarts without first shutting down running processes.
Note: A file system synchronization will not occur if the **-q** flag is used. If you want the file system to be synchronized, manually run the **sync** command or use the **shutdown -r** command.

-t Restarts the system on the specified future date. A valid date has the following format:

mmddHHMM [*yy*]

where:

mm Specifies the month.

dd Specifies the day.

HH Specifies the hour.

MM Specifies the minute.

yy Specifies the year (optional).

Examples

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

```
reboot -l
```

Files

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

Related Information

The **chdev** command, **fasthalt** command, **fsck** command, **halt** command, **lsattr** command, **rc** command, **shutdown** command, **sync** command, **syslogd** command.

The **utmp**, **wtmp**, **failedlogin** File Format.

---

recreatevg Command

Purpose

Recreates a volume group that already exists on a specified set of disks. Imports and varies on the volume group.

Syntax

recreatevg [**-y** *VGname*] [**-p**] [**-f**] [**-Y** *Lv_Prefix* | **-I** *LvNameFile*] [**-L** *Label_Prefix*] [**-n**] *PVname...*

Description

The **recreatevg** command recreates a volume group on a set of disks that are mirrored from another set of disks belonging to a specific volume group. This command allocates new physical volume identifiers (PVID) for the member disks, as the PVIDs are also duplicated by the disk mirroring. Similarly, duplicated logical volume members (LVM) are given new names with the user specified prefixes.

Notes:

1. The **recreatevg** command removes all logical volumes that are fully or partially existed on the physical volumes that are not specified on the command line.
2. The **recreatevg** command gives a warning if the log logical volume for a file system does not exists on the disks specified on the command line.
3. The **recreatevg** command fails if the input list does not match the list compiled from the Volume Group Descriptor Area (VGDA).

Flags

-f	Recreates a volume group from a subset of disks. Only those disks and the logical volumes that reside on this subset of disks will be contained within the recreated volume group. All other disks and logical volumes in the orginal volume group will be deleted from the recreated volume group.
-l *LvNameFile*	Changes logical volume names to the name specified by *LvNameFile*. Entries must be in the format LV:NEWLV1. All logical volumes that are not included in *LvNameFile* are recreated with default system generated names. NEWLV1 name may be the same as LV name in the *LvNameFile* stanza (LV:NEWLV1) to leave the logical volume with the same name.
-L *Label_Prefix*	Changes the labels of logical volumes on the VG being recreated to this prefix. You must modify the /etc/filesystems stanza manually if a simple modification of the mount point is not enough to define the stanza uniquely. Specifying / (slash) as the *Label_Prefix*, will leave the label in the logical volume unchanged.
-n	Specifies that after **recreatevg** the volume group is imported but varied off. Default is imported and vary on.
-p	Disables the automatic generation of the new PVIDs. If the **-p** flag is used, you must ensure that there are no duplicated PVIDs on the system. All the disks that are hardware mirrored must have their PVIDs changed to a unique value.
-y *VGname*	Allows the volume group name to be specified rather than having the name generated automatically. Volume group names must be unique system wide and can range from 1 to 15 characters. The name cannot begin with a prefix already defined in the PdDv class in the Device Configuration database for other devices. The new volume group name is sent to standard output.
-Y *Lv_Prefix*	Causes the logical volumes on the volume group being recreated to be renamed with this prefix. The total length of the prefix and the logical volume name must be less than or equal to 15 characters. If the length exceeds 15 characters, the logical volume is renamed with a default name. The name cannot begin with a prefix already defined in the PdDv class of the Device Configuration Database, nor be a name already used by another device. Specifying NA as the *Lv_Prefix*, will leave all the logical volume names unchanged.

Security

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

Examples

1. To recreate a volume group that contains three physical volumes, type:

   ```
   recreatevg hdisk1 hdisk2 hdisk3
   ```

   The volume group on hdisk1, hdisk2, and hdisk3 is recreated with an automatically generated name, which is displayed.
2. To recreate a volume group on hdisk1 with the new name **testvg**, type:

   ```
   recreatevg -y testvg hdisk1
   ```
3. To recreate a volume group on hdisk14, recreate all logical volumes in that volume group, and rename them with the prefix **newlv**, type:

   ```
   recreatevg -Y newlv hdisk14
   ```

Files

/usr/sbin	Directory where the **recreatevg** command resides.

Related Information

The **chvg** command, **chlv** command.

The Physical and Logical Volume Storage Management Tasks section in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*.

---

redefinevg Command

Purpose

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

Syntax

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

Description

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

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

Flags

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

Example

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

```
redefinevg -d hdisk0 rootvg
```

Files

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

Related Information

The **varyonvg** command, **varyoffvg** command, **lsvg** command.

---

reducevg Command

Purpose

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

Syntax

reducevg [**-d**] [**-f**] *VolumeGroup PhysicalVolume ...*

Description

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

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

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

Notes:

1. To use this command, you must either have root user authority or be a member of the **system** group.
2. Sometimes a disk is removed from the system without first running **reducevg** *VolumeGroup PhysicalVolume*. The VGDA still has this removed disk in it's memory, but the *PhysicalVolume* name no longer exists or has been reassigned. To remove references to this missing disk you can still use **reducevg**, but with the Physical Volume ID (PVID) instead of the disk name: **reducevg** *VolumeGroup PVID*.
3. The **reducevg** command is not allowed on a snapshot volume group.

You can use the Volumes application in Web-based System Manager (wsm) to change volume characteristics.

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

Flags

-d Deallocates the existing logical volume partitions and then deletes resultant empty logical volumes from the specified physical volumes. User confirmation is required unless the **-f** flag is added.

> **Attention:** The **reducevg** command with the **-d** flag automatically deletes all logical volume data on the physical volume before removing the physical volume from the volume group. If a logical volume spans multiple physical volumes, the removal of any of those physical volumes may jeopardize the integrity of the entire logical volume.

-f Removes the requirement for user confirmation when the **-d** flag is used.

Examples

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

   ```
   reducevg vg01 hdisk1
   ```

2. To remove physical volume `hdisk1` and all residing logical volumes from volume group `vg01` without user confirmation, enter the following command. **Attention:** The **reducevg** command with the **-d** flag automatically deletes all logical volume data before removing the physical volume.

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

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

Files

/etc Directory where the **reducevg** command resides.
/tmp Directory where the temporary files are stored and while the command is running.

Related Information

The **extendvg** command, **rmlv** command.

The Logical Volume Storage Overview in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices*.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.2 Web-based System Manager Administration Guide*.

The System Management Interface Tool in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices*.

---

refer Command

Purpose

Finds and inserts literature references in documents.

Syntax

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

Description

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

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

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

> **Note:** Anytime you edit a reference file, you must reissue the **indxbib** command on that file. If you do not use the **indxbib** command, remove any **.ia**, **.ib**, **.ic**, and **.ig** files associated with that reference file; otherwise, you will get a `too many hits` error message from the **refer** command.

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

Flags

-b Bare mode: do not put any flags in text (either numbers or labels).

-e Instead of leaving the references where encountered, accumulates them until a sequence of the following form is encountered:

```
.[
$LIST$
.]
```

then writes out all references collected so far.

-P Places punctuation marks after the reference signal, rather than before. The punctuation marks are locale-specific and are defined in the **refer** message catalog.

-S Produces references in the natural or social science format.

-a *Number* Reverses the first specified number of author names (Jones, J. A. instead of J. A. Jones). If the *Number* variable is omitted, all author names are reversed.

-B *Label***.***Macro* Specifies bibliography mode. Takes a file composed of records separated by blank lines and turns that file into **troff** command input. The specified label is turned into the specified macro, with the *Label* variable value defaulting to **%X** and the *.Macro* variable value defaulting to **.AP** (annotation paragraph).

-c *Keys* Capitalizes, with SMALL CAPS, the fields whose key letters are in the *Keys* variable. For example, Jack becomes JACK .

-f *Number* Sets the footnote number to the specified number instead of the default of 1. With labels rather than numbers, this flag has no effect. See the **-k** flag and the **-l** flag.

-k *Label* Instead of numbering references, uses labels as specified in a reference data line beginning with **%***Label*. By default, the *Label* variable value is **L.**

-l *Letter***,***Digit* Instead of numbering references, uses labels made from the senior author's last name and the year of publication. Only the first specified letters of the last name and the last specified digits of the date are used. If either the *Letter* variable or the *Digit* variable is omitted, the entire name or date, respectively, is used.

-n Does not search the default **/usr/share/dict/papers/Ind** file .If the **REFER** environment variable is set, the specified file is searched instead of the default file. In this case, the **-n** flag has no effect.

-p *Reference* Takes the *Reference* variable as a file of references to be searched. The default file is searched last.

-s *Keys* Sorts references by fields whose key letters are specified by the *Keys* variable string. Renames reference numbers in text accordingly. Implies the **-e** flag. The key letters specified by the *Keys* variable can be followed by a number to indicate how many such fields are used, with q + (plus sign) indicating a very large number. The default value is **AD**, which sorts first by senior author and then by date. For example, to sort on all authors and then title, enter `-sA+T`**.**

It is important to note that blank spaces at the end of lines in bibliography fields cause the records to sort and reverse incorrectly. Sorting large numbers of references can cause a core dump.

Example

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

```
%A M.E. Lesk

%T Some Applications of Inverted Indexes on the UNIXSystem

%B UNIXProgrammer's Manual

%V 2b

%I Bell Laboratories
%C Murray Hill, NJ

%D 1978
```

Files

/usr/share/dict/papers/lnd	Contains the default reference file.
/usr/lbin/refer	Contains companion programs.

Related Information

The **eqn** command, **addbib** command, **indxbib** command, **lookbib** command, **neqn** command, **nroff** command, **roffbib** command, **sortbib** command, **tbl** command, **troff** command.

---

refile Command

Purpose

Moves files between folders.

Syntax

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

Description

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

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

Flags

-draft	Copies the current draft message from your mail directory.
-file *File*	Copies the specified file. The file must be in valid message format. Use the **inc** command to format and file new messages correctly.
+*Folder*	Copies the messages to the specified folder. Any number of folders can be specified.
-help	Lists the command syntax, available switches (toggles), and version information. **Note:** For MH, the name of this flag must be fully spelled out.
-link	Leaves the messages in the source folder or file after they are copied.

Messages Specifies the messages to be copied. You can specify several messages, a range of messages, or a single message. Use the following references to specify messages:

Number
Number of the message.

Sequence
A group of messages specified by the user. Recognized values include:

all All the messages in a folder.

cur or . (period)
Current message. This is the default.

first First message in a folder.

last Last message in a folder.

next Message following the current message.

prev Message preceding the current message.

/DT> If the **-link** and **all** flags are used together, the current message in the current folder does not change. Otherwise, if a message is specified, the refiled message becomes the current message.

-nolink Removes the messages from the source folder or file after they are copied. This flag is the default.

-nopreserve Renumbers the messages that are copied. Renumbering begins with a number one higher than the last message in the destination folder. This flag is the default.

-preserve Preserves the message numbers of copied messages. If messages with these numbers already exist, the **refile** command issues an error message and does not alter the contents of the folders.

-src +*Folder* Identifies the source folder. By default, the system uses the current folder.

Profile Entries

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

`Current-Folder:`	Sets the default current folder.
`Folder-Protect:`	Sets the protection level for your new folder directories.
`Path:`	Specifies the *UserMhDirectory*.
`rmmproc:`	Specifies the program used to remove messages from a folder.

Examples

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

   ```
   refile  +meetings
   ```

   The system responds with a message similar to the following:

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

   Enter `y` to create the folder. A copy of the original message is not retained in the current folder.
2. To copy the current message from the current folder and to the `meetings` folder, enter:

   ```
   refile  -link  +meetings
   ```

   The original message remains in the current folder.

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

   ```
   refile  -draft  +test
   ```

   A copy of the message draft is not retained in the current folder.
4. To refile the current message from the current folder and into several folders, enter:

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

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

Files

$HOME/.mh_profile	Sets the MH user profile.
/usr/bin/refile	Contains the **refile** command.

Related Information

The **folder** command, **folders** command.

The **.mh_alias** file format, **.mh_profile** file format.

Mail Overview in *AIX 5L Version 5.2 System User's Guide: Communications and Networks*.

---

refresh Command

Purpose

Requests a refresh of a subsystem or group of subsystems.

Syntax

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

Description

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

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

Flags

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

Examples

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

   ```
   refresh  -g tcpip
   ```

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

   ```
   refresh  -s xntpd
   ```

Files

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

Related Information

The **startsrc** command, the **stopsrc** command.

The System Resource Controller Overview in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*.

---

refrsrc Command

Purpose

Refreshes the resources within the specified resource class.

Syntax

refrsrc [**–h**] [**–TV**] *resource_class*

Description

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

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

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

Flags

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

–T Writes the command's trace messages to standard error. For your software-service organization's use only.

–V Writes the command's verbose messages to standard output.

Parameters

resource_class Specifies the resource class name.

Security

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

Exit Status

0 Command has run successfully.

1 Error occurred with RMC.

2 Error occurred with CLI script.

3 Incorrect flag on command line.

4 Incorrect parameter on command line.

5 Error occurred with RMC that was based on faulty command line input.

Environment Variables

CT_CONTACT

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

CT_MANAGEMENT_SCOPE

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

0 Specifies *local* scope.

1 Specifies *local* scope.

2 Specifies *peer domain* scope.

3 Specifies *management domain* scope.

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

Standard Output

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

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

Standard Error

All trace messages are written to standard error.

Examples

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

```
refrsrc IBM.FileSystem
```

Location

/usr/sbin/rsct/bin/refrsrc Contains the **refrsrc** command

Related Information

Commands: **lsrsrc**, **lsrsrcdef**

Files: **rmccli**

Books: see the *RSCT Guide and Reference* for information about RMC operations

---

regcmp Command

Purpose

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

Syntax

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

Description

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

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

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

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

Flag

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

Examples

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

   ```
   regcmp stdin1 stdin2
   ```

   This creates the `stdin1.i` and `stdin2.i` files.
2. To creates `stdin1.c` and `stdin2.c` files, enter:

   ```
   regcmp - stdin1 stdin2
   ```

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

File

/usr/ccs/bin/regcmp Contains the **regcmp** command.

Related Information

The **regcmp** or **regex** subroutine.

The Subroutines Overview in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

---

rembak Command

Purpose

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

Syntax

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

Description

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

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

Flags

-# *JobNumber*
: Specifies the *JobNumber* to cancel.

-C
: Sends control file first. The **lpd** protocol allows two handshaking sequences for processing a print job. The default consists of sending the data files(s) first followed by the control file. The other sequence is to send the control file first followed by the data file(s). If **-C** is specified, **rembak** will send the control file first followed by the data file(s).

-D *DebugOutputfile*
: Turns on the debugging option for **rembak**. If no output file name is specified, or if there are any problems creating or writing to the output file, the debugging option is ignored. If the output file specified already exists, new debugging output is appended to the end of it.

-L
: Indicates a long (verbose) status request from the remote queue.

-N *Filter*
: Indicates the machine type of the remote server. The filter name is specified by the **s_statfilter** attribute in the **/etc/qconfig** file. Values for the *filter* variable include the following:

  /usr/lib/lpd/aixshort
  : Indicates the server is another AIX machine.

  /usr/lib/lpd/aixv2short
  : Indicates the server is an RT with an AIX Version 2 operating system.

  /usr/lib/lpd/bsdshort
  : Indicates the server is a bsd machine

  /usr/lib/lpd/attshort
  : Indicates the server is an AT&T machine

-o *Option*
: Specifies an *Option* to be sent to the backend on the remote server. (These *Options* are passed through the **rembak** command.)

-P *Queue*
: Specifies the name of the *Queue* on the remote server where the print job is sent.

-q
: Indicates a short (abbreviated) status request from the remote queue.

-R Restarts the remote queuing system.

Note: The **-R** flag is not supported when sending a request to an operating system. The **lpd** daemon does not support such a request. The **-R** flag is supported only for compatibility with other systems.

-S *Server* Specifies the name of the remote print *Server* where the print request is sent.

-T *Timeout* Sets a timeout period, in minutes, for **rembak** to wait for acknowledgements from the remote server. If no value is specified, a default timeout of 90 seconds is used. This default is also used if Timeout is 0 or a negative value.

-u *UserName@HostName* Cancels a print job for *UserName* that was submitted from the *HostName* machine.

Note: The queuing system does not support multibyte host names.

-X Specifies that the **rembak** command send the **-o** *Option* to the remote server, even if the remote server is a non-AIX machine. If the remote is a non-AIX machine, then the *Option* is sent without the **-o** flag. Thus, **-o -abc** is sent as **-abc**.

To use the **-X** flag on a remote queue, the following line for the specific queue must be included in the **/etc/qconfig** file:

```
backend = /usr/lib/lpd/rembak -X
```

The **qprt**, **lpr** and other queuing commands are not guaranteed to work when -X is specified on a queue. Use the **enq** command.

-x Cancels a job request. Use the **-#** *JobNumber* flag or the **-u** *UserName* flag to cancel a request.

Examples

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

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

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

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

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

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

Files

/usr/lib/lpd/rembak	Contains the **rembak** command.
/etc/hosts.lpd	Contains host names that are allowed to do print requests.
/etc/hosts.equiv	Contains host names that are allowed to do print requests.

Related Information

The **cancel** command, **chque** command, **chquedev** command, **disable** command, **enable** command, **enq** command, **lp** command, **lpd** command, **lpq** command, **lpr** command, **lprm** command, **lpstat** command, **lsallq** command, **lsallqdev** command, **lsque** command, **lsquedev** command, **mkque** command, **mkquedev** command, **qadm** command, **qcan** command, **qchk** command, **qdaemon** command, **qpri** command, **qprt** command, **qstatus** command, **rmque** command, **rmquedev** command.

The **/etc/qconfig** file.

Printer Overview for System Management in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Spooler Overview for System Management in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Remote Printing Overview in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Using Remote Host Access for Printing in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

---

remove Command

Purpose

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

Syntax

/usr/sbin/acct/remove

Description

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

Security

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

Files

/usr/sbin/acct	The path to the accounting commands.
/var/adm/acct/nite	Contains accounting data files.
/var/adm/acct/sum	Cumulative directory for daily accounting records.

Related Information

For more information about the Accounting System, the preparation of daily and monthly reports, and the accounting files, see the Accounting Overview in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*.

Setting Up an Accounting System in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices* explains the steps you must take to establish an accounting system.

---

renice Command

Purpose

Alters the nice value of running processes.

Syntax

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

Description

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

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

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

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

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

Notes:

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

Flags

-g	Interprets all IDs as unsigned decimal integer process group IDs.
-n *Increment*	Specifies the number to add to the nice value of the process. The value of *Increment* can only be a decimal integer from -20 to 20. Positive increment values cause a lower nice value. Negative increment values require appropriate privileges and cause a higher nice value.
-p	Interprets all IDs as unsigned integer process IDs. The -p flag is the default if you specify no other flags.
-u	Interprets all IDs as user name or numerical user IDs.

Exit Status

This command returns the following exit values:

0	Successful completion
>0	An error occurred.

Examples

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

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

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

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

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

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

Files

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

/etc/passwd Maps user names to user IDs.

Related Information

The **nice** command.

The **getpriority** system call, **setpriority** system call.

The Commands Overview in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices* describes how commands are processed.

The Processes Overview in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices* discusses what processes are and input and output redirection.

The Shells Overview in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices* describes what shells are, the different types of shells, and how shells affect the way commands are interpreted.

Controlling Contention for the CPU in *AIX 5L Version 5.2 Performance Management Guide*.

---

reorgvg Command

Purpose

Reorganizes the physical partition allocation for a volume group.

Syntax

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

Description

Attention: This command is not allowed if the volume group is varied on in concurrent mode.

The **reorgvg** command reorganizes the placement of allocated physical partitions within the *VolumeGroup*, according to the allocation characteristics of each logical volume. Use the *LogicalVolume* parameter to reorganize specific logical volumes; highest priority is given to the first logical volume name in the *LogicalVolume* parameter list and lowest priority is given to the last logical volume in the parameter list. The volume group must be varied on and must have free partitions before you can use the **reorgvg** command.

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

Notes:

1. The **reorgvg** command does not reorganize the placement of allocated physical partitions for any striped logical volumes.
2. At least one free physical partition must exist on the specified volume group for the **reorgvg** command to run successfully.
3. To use this command, you must either have root user authority or be a member of the **system** group.
4. If you enter the **reorgvg** command with the volume group name and no other arguments, it will only reorganize the first logical volume in the volume group. The first logical volume is the one listed by the **lsvg -l** *VolumeName* command.

   If you enter the **reorgvg** command with the volume group name and no other arguments, the entire volume group is reorganized.

5. The **reorgvg** command is not allowed on a snapshot volume group or a volume group that has a snapshot volume group.

You can use the Volumes application in Web-based System Manager (wsm) to change volume characteristics.

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

Flags

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

Examples

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

   ```
   reorgvg vg02 lv03 lv04 lv07
   ```

   Only the listed logical volumes are reorganized on `vg02`.
2. To reorganize only the partitions located on physical volumes `hdisk04` and `hdisk06` that belong to logical volumes `lv203` and `lv205`, enter:

   ```
   echo "hdisk04 hdisk06" | reorgvg -i vg02 lv203 lv205
   ```

   The partitions located on physical volumes `hdisk04` and `hdisk06` of volume group `vg02`, that belong to logical volumes `lv203` and `lv205`, are reorganized.

Files

/etc Directory where the **reorgvg** command resides.
/tmp Directory where the temporary files are stored while the command is running.

Related Information

The **chlv** command, **lslv** command, **mklv** command.

The Logical Volume Storage Overview in the *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices*.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.2 Web-based System Manager Administration Guide*.

The System Management Interface Tool in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices*.

Monitoring and Tuning Disk I/O Use in *AIX 5L Version 5.2 Performance Management Guide*.

---

repl Command

Purpose

Replies to a message.

Syntax

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

Description

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

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

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

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

Flags

-annotate	Annotates the message being replied to with the time and date of the reply. You can use the **-inplace** flag to preserve links to an annotated message.
-cc *Names*	Specifies the users who will be listed in the `cc:` field of the reply. You can specify the following variables for *Names*: **all**, **to**, **cc**, and **me**. The default is **-cc all**.
-draftfolder +*Folder*	Places the draft message in the specified folder. If +*Folder* is not specified, then `Current-Folder` is assumed.
-draftmessage *Message*	Specifies the draft message. If you specify **-draftfolder** without the **-draftmessage** flag, the default message is new. If you specify this flag without the **-draftfolder** flag, the system creates the draft in the default file, *UserMHdirectory*/**draft**.
-editor *Editor*	Identifies the initial editor for composing the reply. If you do not specify the **-editor** flag, the **comp** command selects the default editor specified by the `Editor:` entry in your **$HOME/.mh_profile** file.
-fcc +*Folder*	Places a file copy of the reply in the specified folder. If you do not specify this flag, the **repl** command will not produce a file copy.
-filter *File*	Reformats the message being replied to and places the reformatted message in the body of the reply. You must specify a *File* variable with this flag. The **-filter** flag uses the format file acceptable to the **mhl** command.
+*Folder*	Identifies the folder that contains the message to reply to. If a folder is not specified, then `Current-Folder` is used.
-form *FormFile*	Specifies a reply format. The **repl** command treats each line in the specified format file as a format string.
-format	Removes duplicate addresses from the `To:`, `cc:`, and `Bcc:` fields and standardizes these fields using the columns specified by the **-width** flag. The **-format** flag indicates if Internet style is to be used, which serves as the default.

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

Note: For MH, 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 message.

Message Specifies a message. If you specify both a message to reply to and a message draft, you must use the **-draftmessge** flag. Use the following to define a message:

Number
Number of the message.

cur or . (period)
Current message. The default reply message.

first First message in a folder.

last Last message in a folder.

new New message that is created. The default draft message is **new**.

next Message following the current message.

prev Message preceding the current message.

-noannotate Prevents annotation. This flag is the default.

-nocc *Names* Allows you to specify the users who will not be listed in the `cc:` field of the reply. You can specify the following for *Names*: **all**, **to**, **cc**, and **me**.

-nodraftfolder Places the draft in the file *UserMhDirectory*/**draft**.

-noedit Suppresses the initial edit.

-noformat Suppresses both removal of duplicate addresses from the `To:`, `cc:`, and `Bcc:` fields, and standardization of these fields.

-noinplace Prevents annotation in place. This flag is the default.

-noquery Automatically builds the `To:` and `cc:` fields. This flag is the default.

-nowhatnowproc Prevents interactive processing for the **repl** command. This flag prevents editing.

-query Queries you for permission to include each address in the `To:` and `cc:` fields.

-whatnowproc *Program* Starts the specified command string as the program to guide you through the reply tasks. The default is the **whatnow** program.

-width *Number* Sets the width of the address fields. The default is 72 columns.

Profile Entries

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

`Alternate-Mailboxes:` Specifies the mailboxes.

`Current-Folder:` Sets the default current folder.

`Draft-Folder:` Sets the default folder for drafts.

`Editor:` Sets the default editor.

`fileproc:` Specifies the program used to refile messages.

`mhlproc:` Specifies the program used to filter the message for which you are creating a reply.

`Msg-Protect:` Sets the protection level for the new message files.

`Path:` Specifies the user's MH directory.

`whatnowproc:` Specifies the program used to prompt `What now?` questions.

Examples

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

   ```
   repl
   ```

   The system responds with text similar to the following:

```
To: patrick@venus
cc: tom@thomas
Subject: Re: Meeting on Monday
In-reply-to: (Your message of Thu, 21 Jul 88 13:39:34 CST.)
             <8807211839.AA01868>
--------------
```

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

```
What now?
```

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

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

```
repl  +inbox 4
```

The system responds with a message similar to the following:

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

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

```
What now?
```

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

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

```
repl  -annotate
```

The system responds with a message similar to the following:

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

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

```
What now?
```

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

Files

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

Related Information

The **anno** command, **comp** command, **dist** command, **forw** command, **mhl** command, **send** command, **whatnow** command.

The **mh_alias** file format, **mh_profile** file format.

Mail Overview in *AIX 5L Version 5.2 System User's Guide: Communications and Networks*.

---

replacepv Command

Purpose

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

Syntax

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

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

Description

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

Notes:

1. The *DestinationPhysicalVolume* size must be at least the size of the *SourcePhysicalVolume*.
2. The **replacepv** command cannot replace a *SourcePhysicalVolume* with stale logical volume unless this logical volume has a non-stale mirror.
3. The **replacepv** command is not allowed on a snapshot volume group or a volume group that has a snapshot volume group.

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

Flags

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

Security

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

Examples

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

   ```
   replacepv hdisk1 hdisk6
   ```

Files

/usr/sbin	Directory where the **replacepv** command resides.
/tmp	Directory where the temporary files are stored while the command is running.

Related Information

The **migratepv** command.

Logical Volume Storage Overview in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices*.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.2 Web-based System Manager Administration Guide*.

System Management Interface Tool in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices*.

System Dump Facility in *AIX 5L Version 5.2 Kernel Extensions and Device Support Programming Concepts*.

---

repquota Command

Purpose

Summarizes quotas for a file system.

Syntax

repquota [**-v**] [**-g**] [**-u**] { -a | *FileSystem ...* }

Description

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

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

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

Flags

-a	Specifies that quotas are printed for all file systems enabled with quotas in the **/etc/filesystems** file.
-g	Specifies that only group quotas are printed.
-u	Specifies that only user quotas are printed.
-v	Prints a header line before the summary of quotas for each file system.

Security

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

Examples

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

```
repquota -u /u
```

The system prints the following information:

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

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

Files

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

Related Information

The **quota** command, **quotaon** and **quotaoff** commands, **edquota** command, **quotacheck** command.

The Disk Quota System Overview in *AIX 5L Version 5.2 Security Guide*.

Setting Up the Disk Quota System in *AIX 5L Version 5.2 Security Guide*.

---

reset Command

Purpose

Initializes terminals.

Syntax

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

Description

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

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

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

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

Flags

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

Files

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

Related Information

The **csh** command, **sh** command, **stty** command, **tset** command.

The **environ** file, **terminfo** file format.

TERM Values for Different Displays and Terminals section of tty Overview for System Managers in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*.

---

resize Command

Purpose

Sets the **TERMCAP** environment variable and terminal settings to the current window size.

Syntax

resize [**-c** | **-u**] [**-s** [*Rows Columns*]]

Description

The **resize** command utility prints a shell command for setting the **TERM** and **TERMCAP** environment variables to indicate the current size of the xterm window from which the command is run. For this output to take effect, the **resize** command must either be evaluated as part of the command line (usually done with a shell alias or function) or else be redirected to a file that can then be read in. From the C shell (usually known as **/bin/csh**), the following alias could be defined in the user's **.cshrc** file:

```
%  alias rs 'set noglob; `eval resize`'
```

After resizing the window, the user would enter:

```
%  rs
```

Users of versions of the Bourne shell (usually known as **/bin/sh**) that do not have command functions will need to send the output to a temporary file and then read it back in with the . (dot) command:

```
$  resize >/tmp/out
$  . /tmp/out
```

Flags

-c	Indicates that C shell commands should be generated even if the user's current shell is not **/bin/csh**.
-u	Indicates that Bourne shell commands should be generated even if the user's current shell is not a Bourne shell.
-s [*Rows Columns*]	Indicates that Sun console escape sequences will be used instead of the special xterm escape code. If the *Rows* and *Columns* parameters are given, the **resize** command will ask the xterm window to resize itself. However, the window manager may choose to disallow the change.

Note: The -**c** or -**u** must appear to the left of -**s** if both are specified.

File

/etc/termcap	Provides modification for the base termcap entry.

Related Information

The **csh** command, **tset** command, **xterm** command.

---

restart-secldapclntd Command

Purpose

The **restart-secldapclntd** script is used to stop the currently running **secldapclntd** daemon process and then restart it.

Syntax

/usr/sbin/restart-secldapclntd [**-C** *CacheSize*] [**-p** *NumOfThread*] [**-t** *CacheTimeOut*] [**-T** *HeartBeatIntv*]

Description

The **restart-secldapclntd** script stops the **secldapclntd** daemon if it is running, and then restarts it. If the **secldapclntd** daemon is not running, it simply starts it.

Flags

By default, the **secldapclntd** daemon reads the configuration information specified in the **/etc/security/ldap/ldap.cfg** file at startup. If the following options are given in command line when starting **secldapclntd** process, the options from the command line will overwrite the values in the **/etc/security/ldap/ldap.cfg** file.

-C *CacheSize*	Sets the maximum cache entries used by the **secldapclntd** daemon to CacheSize number of entries. Valid range is 100-10,000 entries for user cache. The default is 1000. The group cache entries will be 10% of the user cache entries.
-p *NumOfThread*	Sets the number of thread used by the **secldapclntd** daemon to **NumOfThread** threads. Valid range is 1-1000. The default is 10.
-t *CacheTimeout*	Sets the cache to expire in CacheTimeout seconds. Valid range is 60- 3600 seconds. The default is 300 seconds.
-T *HeartBeatIntv*	Sets the time interval of heartbeat between this client and the LDAP server. Valid values are 60-3,600 seconds. Default is 300.

Examples

1. To restart the **secldapclntd** daemon, type:

   ```
   /usr/sbin/restart-secldapclntd
   ```

2. To restart the **secldapclntd** with using 30 threads and cache timeout value of 500 seconds, type:

   ```
   /usr/sbin/restart-secldapclntd -p 30 -t 500
   ```

Files

/etc/security/ldap/ldap.cfg	Contains information needed by the secldapclntd daemon to connect to the server.

Related Information

The **secldapclntd** daemon

The **mksecldap**, **stop-secldapclntd**, **start-secldapclntd**, **ls-secldapclntd**, and **flush-secldapclntd** commands.

The **/etc/security/ldap/ldap.cfg** file.

---

restbase Command

Purpose

Reads the base-customized information from the boot image, and restores it into the Device Configuration database used during system boot phase 1.

Syntax

restbase [**-o** *File*] [**-d** *Path*] [**-v**]

Description

The **restbase** command reads the base-customized information from the boot disk and puts it in the specified Device Configuration database directory. By default, the base information is read from the boot disk. If no Device Configuration database directory is specified, then the **restbase** command restores this information into the **/etc/objrepos** directory. You can use the **-o** flag to specify a file, other than the boot disk, from which to read the base-customized information.

Attention: The **restbase** command is intended to be executed only during phase 1 of system boot. Do not execute it in a run-time environment or you could destroy the Device Configuration database.

Flags

-o *File*	Specifies a file that contains base-customized data.
-d *Path*	Specifies a directory containing the base Device Configuration database.
-v	Causes verbose output to be written to standard output.

Examples

1. To restore base-customized information and see verbose output, enter:
   ```
   restbase -v
   ```
2. To restore base information into an alternate device database, enter:
   ```
   restbase -d /tmp/objrepos
   ```

Files

/usr/lib/objrepos/PdDv	Contains entries for all known device types supported by the system.
/etc/objrepos/CuDv	Contains entries for all device instances defined in the system.
/etc/objrepos/CuAt	Contains customized device-specific attribute information.
/etc/objrepos/CuDep	Describes device instances that depend on other device instances.
/etc/objrepos/CuDvDr	Stores information about critical resources that need concurrency management through the use of the Device Configuration Library routines.

Related Information

The **bosboot** command, **savebase** command.

Device Configuration Subsystem: Programming Introduction in *AIX 5L Version 5.2 Kernel Extensions and Device Support Programming Concepts*.

Object Data Manager (ODM) Overview for Programmers in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

List of Device Configuration Commands in *AIX 5L Version 5.2 Kernel Extensions and Device Support Programming Concepts*.

---

restore Command

Purpose

Extracts files from archives created with the **backup** command.

Syntax

To Restore Files Archived by File Name

restore -x [**d M v q**] [**-b** *Number*] [**-f** *Device*] [**-s** *SeekBackup*] [**-X** *VolumeNumber*] [*File* ...]

restore -x [**d M v q e**] [**-b** *Number*] [**-f** *Device*] [**-s** *SeekBackup*] [**-X** *VolumeNumber*] [*File* ...]

To List Files Archived by File Name

restore -T [**a q v**] [**-b** *Number*] [**-f** *Device*] [**-s** *SeekBackup*]

To Restore Files Archived by File System
restore -r [**B q v y**] [**-b** *Number*] [**-f** *Device*] [**-s** *SeekBackup*]

To Restore Files Archived by File System
restore -R [**B v y**] [**-b** *Number*] [**-f** *Device*] [**-s** *SeekBackup*]

To Restore Files Archived by File System
restore -i [**h m q v y**] [**-b** *Number*] [**-f** *Device*] [**-s** *SeekBackup*]

To Restore Files Archived by File System
restore -x [**B h m q v y**] [**-b** *Number*] [**-f** *Device*] [**-s** *SeekBackup*] [*File* ...]

To List Files Archived by File System
restore -t | **-T** [**Bah q v y**] [**-b** *Number*] [**-f** *Device*] [**-s** *SeekBackup*] [*File* ...]

To Restore File Attributes Archived by File Name
restore -Pstring [**B d qv**] [**b***Number*] [**s** *SeekNumber*] [**-f** *Device*] [*File* ...]

To Restore File Attributes Archived by File System
restore -Pstring [**hqv**] [**b** *Number*] [**s** *SeekNumber*] [**-f** *Device*] [*File* ...]

Description
The **restore** command reads archives created by the **backup** command and extracts the files stored on them. These archives can be in either file-name or file-system format. An archive can be stored on disk, diskette, or tape. Files must be restored using the same method by which they were archived. This requires that you know the format of the archive. The archive format can be determined by examining the archive volume header information that is displayed when using the **-T** flag. When using the **-x**, **-r**, **-T**, or **-t** flags, the **restore** command automatically determines the archive format.

> **Note: restore** actively sparses files that are being restored. If a file has block aligned and sized areas that are NULL populated, then **restore**does not cause physical space for those filesystem blocks to be allocated. The size in bytes of the file remain the same, but the actual space taken within the filesystem is only for the non-NULL areas.

Individual files can be restored from either file-name or file-system archives by using the **-x** flag and specifying the file name. The file name must be specified as it exists on the archive. Files can be restored interactively from file-system archives using the **-i** flag. The names of the files on an archive can be written to standard output using the **-T** flag.

Users must have write access to the file system device or have Restore authorization in order to extract the contents of the archive.

The diskette device, **/dev/rfd0**, is the default media for the **restore** command. To restore from standard input, specify a **-** (dash) with the **-f** flag. You can also specify a range of devices, such as **/dev/rmt0-2**.

> **Notes:**
>
> 1. If you are restoring from a multiple-volume archive, the **restore** command reads the volume mounted, prompts you for the next volume, and waits for your response. After inserting the next volume, press the Enter key to continue restoring files.
> 2. If an archive, created using the **backup** command, is made to a tape device with the device block size set to 0, it may be necessary for you to have explicit knowledge of the block size that was used when the tape was created in order to restore from the tape.
> 3. Multiple archives can exist on a single tape. When restoring multiple archives from tape, the **restore** command expects the input device to be a no-retension-on-open, no-rewind-on-close tape device. Do not use a no-rewind tape device for restoring unless either the **-B**, **-s**, or **-X** flag is specified. For more information on using tape devices, see the **rmt** special file.

File-System Archives
File-system archives are also known as i-node archives due to the method used to archive the files. A file-system name is specified with the **backup** command, and the files within that file system are archived based on their structure and layout within the file system. The **restore** command restores the files on a file-system archive without any special understanding of the underlying structure of the file system.

When restoring file-system archives, the **restore** command creates and uses a file named **restoresymtable**. This file is created in the current directory. The file is necessary for the **restore** command to do incremental file-system restores.

> **Note:** Do not remove the **restoresymtable** file if you perform incremental file-system backups and restores.

The *File* parameter is ignored when using either the **-r** or the **-R** flag.

File-Name Archives
File-name archives are created by specifying a list of file names to archive to the **backup** command. The **restore** command restores the files from a file-name archive without any special understanding of the underlying structure of the file system. The **restore** command allows for metacharacters to be used when specifying files for archive extraction. This provides the capability to extract files from an archive based on pattern matching. A pattern filename should be enclosed in single quotations, and patterns should be enclosed in brackets (...).

About Sparse Files
Operating system filesystems, files containing long strings of NULLs can be stored much more efficiently than other files. To be specific, if a string of NULLs spans an entire allocation block, that whole block is not stored on disk at all. Files where one or more blocks are omitted in this way are called sparse files. The missing blocks are also known as holes.

> **Note :**Sparse files are not the same as compressed files. Sparse files are exactly the same as their non-sparse equivalents when they are read.

Sparse files are generally created by database applications. Since whenever a database file is created it is formatted with NULLs. These fragments may be either filled with allocated or unallocated NULLs.

Flags

-a Specified with the **t** and/or **T** option, the **-a** option displays the list of files in the archive, along with their permissions.

-B Specifies that the archive should be read from standard input. Normally, the **restore** command examines the actual medium to determine the backup format. When using a | (pipe), this examination cannot occur. As a result, the archive is assumed to be in file-system format, and the device is assumed to be standard input (**-f -**).

-b *Number*

For backups done by name, specifies the number of 512-byte blocks; for backups done by i-node, specifies the number of 1024-byte blocks to read in a single output operation. When the **restore** command reads from tape devices, the default is 100 for backups by name and 32 for backups by i-node.

The read size is the number of blocks multiplied by the block size. The default read size for the **restore** command reading from tape devices is 51200 (100 * 512) for backups by name and 32768 (32 * 1024) for backups by i-node. The read size must be an even multiple of the tapes physical block size. If the read size is not an even multiple of the tapes physical block size and it is in fixed block mode (nonzero), the **restore** command tries to determine a valid value for *Number*. If successful, the **restore** command changes *Number* to the new value, writes a message about the change to standard output, and continues. If unsuccessful in finding a valid value for *Number*, the **restore** command writes an error message to standard error and exits with a nonzero return code. Larger values for the *Number* parameter result in larger physical transfers from the tape device.

The value of the **-b** flag is always ignored when the **restore** command reads from diskette. In this case, the command always reads in clusters that occupy a complete track.

-d

Indicates that, if the *File* parameter is a directory, all files in that directory should be restored. This flag can only be used when the archive is in file-name format.

-e

Restores the non sparse files as non sparse as they were archived by the name format of **backup** command for both packed and unpacked files. It is necessary to know the sparseness/non-sparseness of the file(s) before archiving the files, since enabling this flag restores the sparse files as non-sparse.

This flag should be enabled only if files are to be restored are non sparse consisting of more than 4K NULLs. If the **-e** flag is specified during **restore**, it successfully restores all normal files normally and non-sparse database files as non sparse.

-E

The **-E** option requires one of the following arguments. If you omit the **-E** option, **warn** is the default behavior.

force Fails the restore operation on a file if the fixed extent size or space reservation of the file cannot be preserved.

ignore Ignores any errors in preserving extent attributes.

warn Issues a warning if the space reservation or the fixed size of the file cannot be preserved.

-f *Device*

Specifies the input device. To receive input from a named device, specify the *Device* variable as a path name (such as `/dev/rmt0`). To receive input from the standard output device, specify a - (minus sign). The - (minus) feature enables you to pipe the input of the **restore** command from the **dd** command.

You can also specify a range of archive devices. The range specification must be in the following format:

```
/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 **restore** command automatically goes from one device in the range to the next. After exhausting all of the specified devices, the **restore** command halts and requests that new volumes be mounted on the range of devices.

-h Restores only the actual directory, not the files contained in it. This flag can only be used when the archive is in file-system format. This flag is ignored when used with the **-r** or **-R** flags.

-i Allows you to interactively restore selected files from a file-system archive. The subcommands for the **-i** flag are:

cd *Directory*
Changes the current directory to the specified directory.

add [*File***]**
Specifies that the *File* parameter is added to the list of files to extract. If *File* is a directory, that directory and all the files contained in it are added to the extraction list (unless the **-h** flag is used). If *File* is not specified, the current directory is added to the extraction list.

delete [*File***]**
Specifies that the *File* parameter is to be removed from the list of files to be extracted. If *File* is a directory, that directory and all the files contained in it are removed from the extraction list (unless the **-h** flag is used).

ls [*Directory***]**
Displays the directories and files contained within the *Directory* parameter. Directory names are displayed with a / (slash) after the name. Files and directories, within the specified directory, that are on the extraction list are displayed with an * (asterisk) before the name. If verbose mode is on, the i-node number of the files and directories is also displayed. If the *Directory* parameter is not specified, the current directory is used.

extract Restores all the directories and files on the extraction list.

pwd Displays the full path name of the current directory.

verbose
Causes the **ls** subcommand to display the i-node number of files and directories. Additional information about each file is also displayed as it is extracted from the archive.

setmodes
Sets the owner, mode, and time for all directories added to the extraction list.

quit Causes **restore** to exit immediately. Any files on the extraction list are not restored.

help Displays a summary of the subcommands.

-M Sets the access and modification times of restored files to the time of restoration. If a restored file is an archive created by the **ar** command, the modification times in all the member headers are also set to the time of restoration. You can specify the **-M** flag only when you are restoring individually named files and only if the **-x** or **-X** flags are also specified. When the **-M** flag is not specified, the **restore** command maintains the access and modification times as they appear on the backup medium.

The **-M** flag is used when the data is in the AIX 4.2 backup by-i-node or by-name format.

-m Renames restored files to the file's i-node number as it exists on the archive. This is useful if a few files are being restored and you want these files restored under a different file name. Since any restored archive members are renamed to their i-node numbers, directory hierarchies and links are not preserved. Directories and hard links are restored as regular files. The **-m** flag is used when the archive is in file-system format.

-Pstring Restore only the file attributes. Does not restore the file contents. If the file specified does not exist in the target directory path, the files is not created. This flag restores file attributes selectively depending on the flags specified in the string parameter. String parameter can be a combination of the following characters:

A restore all attributes.

a restore only the permissions of the files.

o restore only the ownership of the files.

t restore only the timestamp of the files.

c restore only the ACL attributes of the files.

Note: Among the existing options for the **restore** command, options **v**, **h**, **b**, **s**, **f**, **B**, **d,** and **q** are valid with the **P** option. The **P** option can be used with both file-name and filesystem archives. If the **File** argument is a symbolic link, then the metadata of the target file will be modified and not that of the symbolic link.

Warning: Usage of **-P** flag will overwrite the attributes of files owned by another user when executed by the superuser.

-q Specifies that the first volume is ready to use and that the **restore** command should not prompt you to `mount the volume and hit Enter`. If the archive spans multiple volumes, the **restore** command prompts you for the subsequent volumes.

-r Restores all files in a file-system archive. The **-r** flag is only used to restore complete level 0 backups or to restore incremental backups after a level 0 backup is restored. The **restoresymtable** file is used by **restore** to pass information between incremental restores. This file should be removed once the last incremental backup is restored. The *File* parameter is ignored when using the **-r** flag.

-R Requests a specific volume of a multiple-volume, file-system archive. The **-R** flag allows a previously interrupted restore to be restarted. The *File* parameter is ignored when using the **-R** flag. Once restarted, the **restore** command behavior is the same as with the **-r** flag.

-s *SeekBackup* Specifies the backup to seek and restore on a multiple-backup tape archive. The **-s** flag is only applicable when the archive is written to a tape device. To use the **-s** flag properly, a no-rewind-on-close and no-retension-on-open tape device, such as **/dev/rmt0.1** or **/dev/rmt0.5**, must be specified. If the **-s** flag is specified with a rewind tape device, the **restore** command displays an error message and exits with a nonzero return code. If a no-rewind tape device is used and the **-s** flag is not specified, a default value of **-s 1** is used. The value of the *SeekBackup* parameter must be in the range of 1 to 100 inclusive. It is necessary to use a no-rewind-on-close, no-retension-on-open tape device because of the behavior of the **-s** flag. The value specified with **-s** is relative to the position of the tapes read/write head and not to an archives position on the tape. For example, to restore the first, second, and fourth backups from a multiple-backup tape archive, the respective values for the **-s** flag would be **-s 1**, **-s 1**, and **-s 2**.

-t Displays information about the backup archive. If the archive is in file-system format, a list of files found on the archive is written to standard output. The name of each file is preceded by the i-node number of the file as it exists on the archive. The file names displayed are relative to the root (/) directory of the file system that was backed up. If the *File* parameter is not specified, all the files on the archive are listed. If the *File* parameter is used, then just that file is listed. If the *File* parameter refers to a directory, all the files contained in that directory are listed. If the archive is in file-name format, information contained in the volume header is written to standard error. This flag can be used to determine if the archive is in file-name or file-system format.

-T Displays information about the backup archive. If the archive is in file-name format, the information contained in the volume header is written to standard error, and a list of files found on the archive is written to standard output. The *File* parameter is ignored for file-name archives. If the archive is in file-system format, the behavior is identical to the **-t** flag.

-v Displays additional information when restoring. If the archive is in file-name format and either the **-x** or **-T** flag is specified, the size of the file as it exists on the archive is displayed in bytes. Directory, block, or character device files are archived with a size of 0. Symbolic links are listed with the size of the symbolic link. Hard links are listed with the size of the file, which is how they are archived. Once the archive is read, a total of these sizes is displayed. If the archive is in file-system format, directory and nondirectory archive members are distinguished.

-x Restores individually named files specified by the *File* parameter. If the *File* parameter is not specified, all the archive members are restored. If the *File* parameter is a directory and the archive is in file-name format, only the directory is restored. If the *File* parameter is a directory and the archive is in file-system format, all the files contained in the directory are restored. The file names specified by the *File* parameter must be the same as the names shown by the **restore -T** command. Files are restored with the same name they were archived with. If the file name was archived using a relative path name (**./filename**), the file is restored relative to the current directory. If the archive is in file-system format, files are restored relative to the current directory.

The **restore** command automatically creates any needed directories. When using this flag to restore file-system backups, you are prompted to enter the beginning volume number.

The **restore** command allows for shell-style pattern matching metacharacters to be used when specifying files for archive extraction. The rules for matching metacharacters are the same as those used in shell pathname ″globbing,″ namely:

*** (asterisk)**
Matches zero or more characters, but not a '.' (period) or '/' (slash).

? (question mark)
Matches any single character, but not a '.' (period) or '/' (slash).

[ ] (brackets)
Matches any one of the characters enclosed within the brackets. If a pair of characters separated by a dash are contained within the brackets, the pattern matches any character that lexically falls between the two characters in the current local. Additionally, a '.' (period) or a '/' (slash) within the brackets will not match a a '.' (period) or a '/' (slash) in a file name.

\ (backslash)
Matches the immediately following character, preventing its possible interpretation as a metacharacter.

-X *VolumeNumber* Begins restoring from the specified volume of a multiple-volume, file-name backup. Once started, the **restore** command behavior is the same as with the **-x** flag. The **-X** flag applies to file-name archives only.

-y Continues restoring when tape errors are encountered. Normally, the **restore** command asks you whether or not to continue. In either case, all data in the read buffer is replaced with zeroes. The **-y** flag applies only when the archive is in file-system format.

-? Displays a usage message.

Exit Status

This command returns the following exit values:

0 Successful completion.
>0 An error occurred.

Examples

1. To list the names of files in either a file-name or file-system archive on the diskette device **/dev/rfd0**, type:

```
restore -Tq
```

The archive is read from the **/dev/rfd0** default restore device. The names of all the files and directories contained in the archive are displayed. For file-system archives, the file name is preceded by the i-node number of the file as it exists on the archive. The **-q** flag tells the **restore** command that the first volume is available and is ready to be read. As a result, you are not prompted to mount the first volume.

2. To restore a specific file, type:

   ```
   restore -xvqf myhome.bkup system.data
   ```

   This command extracts the file `system.data` into the current directory from the archive `myhome.bkup`. The archive in this example is in the current directory. File and directory names must be specified as they are displayed when using the **-T** flag. The **-v** flag displays additional information during the extraction. This example applies to both file-name and file-system archives.

3. To restore a specific directory and the contents of that directory from a file-name archive, type:

   ```
   restore -xdvqf /dev/rmt0 /home/mike/tools
   ```

   The **-x** flag tells **restore** to extract files by their file name. The **-d** tells **restore** to extract all the files and subdirectories in the `/home/mike/tools` directory. File and directory names must be specified as they are displayed when using the **-T** flag. If the directories do not exist, they are created.

4. To restore a specific directory and the contents of that directory from a file-system archive, type:

   ```
   restore -xvqf /dev/rmt0 /home/mike/tools
   ```

   This command extracts files by file name. File and directory names must be specified as they are displayed when using the **-T** flag. If the directories do not exist, they are created.

5. To restore an entire file-system archive, type:

   ```
   restore -rvqf /dev/rmt0
   ```

   This command restores the entire file system archived on the tape device, `/dev/rmt0`, into the current directory. This example assumes you are in the root directory of the file system to be restored. If the archive is part of a set of incremental file-system archives, the archives should be restored in increasing backup-level order beginning with level 0 (for example, `0`, `1`, `2`).

6. To restore the fifth and ninth backups from a single-volume, multiple-backup tape, type:

   ```
   restore -xvqs 5 -f/dev/rmt0.1
   restore -xvqs 4 -f/dev/rmt0.1
   ```

   The first command extracts all files from the fifth archive on the multiple-backup tape specified by `/dev/rmt0.1`. The `.1` designator specifies the tape device will not be retensioned when it is opened and that it will not be rewound when it is closed. It is necessary to use a no-rewind-on-close, no-retension-on-open tape device because of the behavior of the **-s** flag. The second command extracts all the files from the fourth archive (relative to the current location of the tape head on the tape). After the fifth archive has been restored, the tape read/write head is in a position to read the archive. Since you want to extract the ninth archive on the tape, you must specify a value of 4 with the **-s** flag. This is because the **-s** flag is relative to your position on the tape and not to an archives position on the tape. The ninth archive is the fourth archive from your current position on the tape.

7. To restore the fourth backup, which begins on the sixth tape on a 10-tape multiple-backup archive, put the sixth tape into the tape drive and type:

   ```
   restore -xcs 2 -f /dev/rmt0.1 /home/mike/manual/chap3
   ```

   Assuming the fourth backup is the second backup on the sixth tape, specifying `-s 2` advances the tape head to the beginning of the second backup on this tape. The **restore** command then restores the specified file from the archive. If the backup continues onto subsequent volumes and the file has not been restored, the **restore** command instructs you to insert the next volume until the end of the backup is reached. The **-f** flag specifies the no-rewind, no-retension tape device name.

Note: The **-s** flag specifies the backup number relative to the tape inserted in the tape drive, not to the overall 10-tape archive.

8. To improve the performance on streaming tape devices, pipe the **dd** command to the **restore** command by typing:

   ```
   dd if=/dev/rmt0 bs=64b | restore -xf- -b64
   ```

   The **dd** command reads the archive from the tape using a block size of 64 512-byte blocks and writes the archive to standard output. The **restore** command reads the standard input using a block size of 64 512-byte blocks. The value of the block size used by the **dd** command to read the archive from the tape must be an even multiple of the block size that was used to create the tape with the **backup** command. For example, the following **backup** command could be used to create the archive that this example extracts:

   ```
   find /home -print | backup -ivqf/dev/rmt0 -b64
   ```

   This example applies to archives in file-name format only. If the archive was in file-system format, the **restore** command should include the **-B** flag.

9. To improve the performance of the **restore** command on the 9348 Magnetic Tape Unit Model 12, you can change the block size by typing:

   ```
   chdev -l DeviceName -a BlockSize=32k
   ```

10. To restore non sparse database files, type:

    ```
    restore  -xef  /dev/rmt0
    ```

11. To restore files that were sparse before archive as sparse, type:

    ```
    restore  -xf  /dev/rmt0
    ```

12. To restore only the permissions of the files from the archive, type:

    ```
    restore -Pa -vf /dev/rmt0
    ```

13. To restore only the ACL attributes of the files from the archive, type:

    ```
     restore -Pc -vf /dev/rmt0
    ```

14. To view the table of contents along with the file permissions, type:

    ```
    restore -Ta -vf /dev/rmt0
    ```

Files

/dev/rfd0	Specifies the default restore device.
/usr/sbin/restore	Contains the **restore** command.

Related Information

The **ar** command, **backup** command, **mkfs** command, **fsck** command, **dd** command.

The **filesystems** file, **backup** file, **rmt** special file.

The Backup Overview for System Management in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*.

The File Systems in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices* explains file system types, management, structure, and maintenance.

The System Management Interface Tool in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices*.

restorevgfiles Command

Purpose

Restores files from a backup source.

Syntax

restorevgfiles [**-b** *blocks*] [**-f** *device*] [**-a**] [**-s**] [**-d** *path*] [**-D**] [*file_list*]

Description

The **restorevgfiles** command restores files from tape, file, CD-ROM, or their volume group backup source. The **restorevgfiles** command also works for multi-volume backups such as multiple CDs, DVDs, or tapes.

The **restorevgfiles** and **listvgbackup -r** commands perform identical operations and should be considered interchangeable. The **restorevgfiles** command automatically applies the **-r** flag. The **-r** flag, while redundant, is retained for compatibility purposes and will cause no unusual behavior if specified. For a complete description of the **-r** flag, see the **listvgbackup** command.

Flags

-b *blocks*	Specifies the number of 512-byte blocks to read in a single input operation, as defined by the *blocks* parameter. If the *blocks* parameter is not specified, the number of blocks read will default to 100.
-f *device*	Specifies the type of device containing the backup (file, tape, CD-ROM, or other source) as defined by the *device* parameter. When **-f** is not specified, *device* will default to **/dev/rmt0**.
-a	Verifies the physical block size of the tape backup, as specified by the **-b** *block* flag. You may need to alter the block size if necessary to read the backup. The **-a** flag is valid only when a tape backup is used.
-s	Specifies that the backup source is a user volume group and not **rootvg**.
-d *path*	Specifies the directory path to which the files will be restored, as defined by the *path* parameter. If the **-d** parameter is not used, the current working directory is used. This can be a problem if the current working directory is root. We recommend writing to a temporary folder instead of to root.
-D	Produces debug output.

Parameters

file_list	Identifies the list of files to be restored. The full path of the files relative to the current directory should be specified in the space-separated list. All files in the specified directory will be restored unless otherwise directed. If you are restoring all files in a directory, we recommend writing to a temporary folder instead of to root.

Examples

1. To read the backup stored at **/dev/cd1** and restore all files to the **/data/myfiles** directory, enter:

   ```
   restorevgfiles -f /dev/cd1 -s -d /data/myfiles
   ```

2. To read the user vg backup from the default device at 20 512-byte blocks at a time and restore the **/myapp/app.h** file to the current directory, enter:

   ```
   restorevgfiles -b 20 -s .myapp/app.h
   ```

3. To read the backup stored at **/dev/cd1** and restore the **/myapp/app.c** file to the **/data/testcode** directory, enter:

   ```
   restorevgfiles -f /dev/cd1 -s -d /data/testcode ./myapp/app.c
   ```

Files

/usr/bin/restorevgfiles	Contains the **restorevgfiles** command

Related Information

The **listvgbackup** command.

---

restvg Command

Purpose

Restores the user volume group and all its containers and files.

Syntax

restvg [**-b** *Blocks*] [**-d** *FileName*][**-f** *Device*] [**-l**] [**-q**] [**-r**] [**-s**] [**-n**] [**-p** *PPsize*] [*DiskName ...*]

Description

The **restvg** command restores the user volume group and all its containers and files, as specified in the **/tmp/vgdata/***vgname***/***vgname***.data** file (where *vgname* is the name of the volume group) contained within the backup image created by the **savevg** command.

The **restvg** command restores a user volume group. The **bosinstall** routine reinstalls the root volume group (**rootvg**). If the **restvg** command encounters a **rootvg** volume group in the backup image, the **restvg** command exits with an error.

If a **yes** value has been specified in the `EXACT_FIT` field of the **logical_volume_policy** stanza of the **/tmp/vgdata/***vgname***/***vgname***.data** file, the **restvg** command uses the map files to preserve the placement of the physical partitions for each logical volume. The target disks must be of the same size or larger then the source disks specified in the **source_disk_data** stanzas of the *vgname***.data** file.

Note: To view the files in the backup image or to restore individual files from the backup image, the user must use the **restore** command with the **-T** or **-x** flag, respectively. (Refer to the **restore** command for more information.)

Flags

-b *Blocks*	Specifies the number of 512-byte blocks to read in a single input operation. If this parameter is not specified the default of 100 is used by the **restore** command. Larger values result in larger physical transfers to tape devices.
DiskName...	Specifies the names of disk devices to be used instead of the disk devices listed in the *vgname***.data** file. Target disk devices must be defined as empty physical volumes; that is, they must contain a physical volume identifier and must not belong to a volume group. If the target disk devices are new, they must be added to the system using the **mkdev** command. If the target disk devices belong to a volume group, they must be removed from the volume group using the **reducevg** command.
-d *FileName*	The -d flag is an optional flag, which, if specified, must be followed by a filename. This file will be used as the **vgname.data** file instead of the one contained within the backup image being restored. The filename can be specified by either a relative or an absolute pathname.
-f *Device*	Specifies the device name of the backup media. The default is **/dev/rmt0**.

-l Displays useful information about a volume group backup.

This flag requires the **-f** *device* flag. This flag causes **restvg** to display information such as volume group, date and time backup was made, uname output from backed up system, oslevel, recommended maintenance level, backup size in megabytes, and backup shrink size in megabytes. The shrink size is the size of the data on all filesystems. The full size is the total size of each filesystem (unused + data). The **-l** flag also displays the logical volume and filesystem information of the backed up volume group, equivalent to running ″**lsvg -l** *vgname*″.

-n Specifies that the existing MAP files are ignored. The **-n** flag overrides the value of the `EXACT_FIT` field in the logical_volume_policy stanza of the *vgname***.data** file.

-p *PPsize* Specifies the number of megabytes in each physical partition. If not specified, **restvg** uses the best value for the *PPsize*, dependent upon the largest disk being restored to. If this is not the same as the size specified in the *vgname***.data** file, the number of partitions in each logical volume will be appropriately altered with respect to the new *PPsize*.

If a *PPsize* is specified that is smaller than appropriate for the disk sizes, the larger *PPsize* will be used.

If a *PPsize* is specified that is larger than appropriate for the disk sizes, the specified larger *PPsize* will be used.

-q Specifies that the usual prompt not be displayed before the restoration of the volume group image. If this flag is not specified, the prompt displays the volume group name and the target disk-device names.

-r Recreates a volume groups structure only. This allows restvg to create (for the specified backup *FileName* or *Device*) the volume group, logical volumes, and filesystems, from the backup, without restoring any files or data. This is useful for users who use third party software for restoring data and just need all the AIX logical volume structure in place.
Note: be used with either the **-f** *Device* flag or the **-d** *FileName* flag. This is because **restvg** requires a backup image or *vgname***.data** file to get all the information it needs to recreate the logical volume structure of the volume group desired.

-s Specifies that the logical volumes be created at the minimum size possible to accommodate the file systems. This size is specified by the value of `LV_MIN_LPS` field of the **lv_data** stanza of the *vgname***.data** file (where *vgname* is the name of the volume group).

The **-s** flag overrides the values of the `SHRINK` and `EXACT_FIT` fields in the **logical_volume_policy** stanza of the *vgname***.data** file. The **-s** flag causes the same effect as values of `SHRINK=yes` and `EXACT_FIT=no` would cause.

Examples

1. To restore the volume group image from the **/dev/rmt1** device, onto the **hdisk2** and **hdisk3** disks, enter:

   ```
   restvg -f/dev/rmt1 hdisk2 hdisk3
   ```

2. To restore the volume group image saved in **/mydata/myvg** file onto the disks specified in the *vgname***.data** file contained within the backup image, enter:

   ```
   restvg -f/mydata/myvg
   ```

3. To recreate the volume group logical volume structure without restoring any files using only the *vgname***.data** file **/home/my_dir/my_vg.data**, enter:

```
restvg -r -d /home/my_dir/my_vg.data
```

Note: *vgname***.data** files can be created for a volume group by using the **mkvgdata** command.

4. To recreate the volume group logical volume structure without restoring any files using the *vgname***.data** file inside of the volume group backup located on the tape in **/dev/rmt0**, enter the following:

   ```
   restvg -r -f /dev/rmt0
   ```

5. To display volume group information about the volume group backed up on the tape in **/dev/rmt0**, enter:

   ```
   restvg -l -f /dev/rmt0
   ```

Related Information

The mkvgdata command, **restore** command, **savevg** command, **mkdev** command, **reducevg** command.

---

rev Command

Purpose

Reverses characters in each line of a file.

Syntax

rev [*File ...*]

Description

The **rev** command copies the named files to standard output, reversing the order of characters in every line. If you do not specify a file, the **rev** command reads standard input.

Examples

To reverse characters in each line of a file, type:

```
rev file
```

If the `file` file contains the text:

```
abcdefghi
123456789
```

then the **rev** command displays:

```
ihgfedcba
987654321
```

Files

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

Related Information

Input and Output Redirection Overview in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices*.

revnetgroup Command

Purpose

Reverses the listing of users and hosts in network group files in NIS maps.

Syntax

/usr/sbin/revnetgroup { **-h** | **-u** } [*File*]

Description

The **revnetgroup** command reverses the order in which hosts and users are listed in the **/etc/netgroup** file. The **revnetgroup** command is called from the **/var/yp/Makefile** file to produce output for creating either the **netgroup.byuser** or **netgroup.byhost** NIS map. Each line in the output file begins with a key formed by concatenating the host or user name with the domain name. Following the key is a list of groups to which the host or user belongs. The list is preceded by a tab, and each group is separated by a comma.

> **Note:** The list of groups does not use the names of universal groups (groups that include all users in the network). Universal groups are listed under * (asterisk).

The **revnetgroup** command takes an optional file name if the default **/etc/netgroup** file is not desired. This feature provides users with flexibility to create custom network group maps.

Flags

-h Produces output for creating the **netgroup.byhost** map.
-u Produces output for creating the **netgroup.byuser** map.

Examples

1. To cause the **/etc/netgroup** file to list user names before host names, modify the appropriate stanza in the **/var/yp/Makefile** to read:

   ```
   revnetgroup -u
   ```

2. To create a new network group file, called `newgroup`, in the **/etc** directory, modify the appropriate stanza in the **/var/yp/Makefile** to read:

   ```
   revnetgroup -h newgroup
   ```

   The **-h** flag used in this example causes the new **/etc/newgroup** file to list host names before user names.

Files

/etc/netgroup	Contains lists of users and hosts in network groups.
/var/yp/Makefile	Contains rules for making NIS maps.

Related Information

The **makedbm** command, **ypinit** command, **yppush** command.

Network File System (NFS) Overview for System Management in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

Network Information Service (NIS) in *AIX 5L Version 5.2 Network Information Services (NIS and NIS+) Guide*.

NIS Reference.

---

rexd Daemon

Purpose

Executes programs for remote machines.

Syntax

/usr/sbin/rpc.rexd

Description

The **rexd** daemon executes programs for remote machines when a client issues a request to execute a program on a remote machine. The **inetd** daemon starts the **rexd** daemon from the **/etc/inetd.conf** file.

Noninteractive programs use standard file descriptors connected directly to TCP connections. Interactive programs use pseudo-terminals, similar to the login sessions provided by the **rlogin** command. The **rexd** daemon can use the network file system (NFS) to mount the file systems specified in the remote execution request. Diagnostic messages are normally printed on the console and returned to the requester.

Note: A root user cannot execute commands using **rexd** client programs such as the **on** command.

Files

/tmp_rex/rexd	Contains temporary mount points for remote file systems.
/etc/exports	Lists the directories that the server can export.
inetd.conf	Starts RPC daemons and other TCP/IP daemons.
/etc/passwd	Contains an entry for each user that has permission to log in to the machine.

Related Information

The **on** command.

The **inetd** daemon.

Network File System (NFS) Overview for System Management in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

List of NFS Commands.

---

rexec Command

Purpose

Executes commands one at a time on a remote host.

Syntax

rexec [**-a**][**-d** | **-n**] [**-i**] *Host Command*

Description

The **/usr/bin/rexec** command executes a command on the specified remote host.

The **rexec** command provides an automatic login feature by checking for a **$HOME/.netrc** file that contains the user name and password to use at the remote host. If such an entry is not found or if your system is operating in secure mode (see the **securetcpip** command), the **rexec** command prompts for a valid user name and password for the remote host. In both cases, **rexec** causes **rexecd** on the remote system to use the default `compat` login authentication method for the user. **rexecd** does not look at the **/etc/security/user** file on the remote system for alternative authentication methods. You can also override the automatic login feature by specifying the **-n** flag on the **rexec** command line.

Flags

-a Indicates the standard error of the remote command is the same as standard output. No provision is made for sending arbitrary signals to the remote process.
-d Enables socket-level debugging.
-i Prevents reading the stdin.
-n Prevents automatic login. With the **-n** flag specified, the **rexec** command prompts for a user name and password to use at the remote host, rather than searching for a **$HOME/.netrc** file.

Parameters

Command Specifies the command, including any flags or parameters, to be executed on the remote host.
Host Specifies in alphanumeric form the name of the host where the command is to be executed.

Examples

1. To execute the **date** command on a remote host, enter:

   ```
   rexec host1 date
   ```

   The output from the date command is now displayed on the local system. In this example, the **$HOME/.netrc** file on the local host contains a user name and password valid at the remote host.

   If you do not have a valid entry in the **$HOME/.netrc** file for the remote host, you will be prompted for your login ID and password. After you have entered the requested login information, the output from the date command is displayed on the local system.
2. To override the automatic login feature and execute the **date** command on a remote host, enter:

   ```
   rexec -nhost1 date
   ```

   Enter your name and password when prompted.

   The output from the date command is now displayed on the local system.
3. To list the directory of another user on a remote host, enter:

   ```
   rexec host1 ls -l /home/karen
   ```

   The directory listing of user `karen` on remote host `host1` is displayed on the local system.

   If you do not have a valid entry in the **$HOME/.netrc** file for the remote host, you will be prompted for your login id and password. After you have entered the requested login information, the directory listing of user `karen` on remote host `host1` is displayed on the local system.

Related Information

Network Overview in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

The **ftp** command, **rlogin** command, **rsh** command, **securetcpip** command.

The **rexecd** daemon.

The **.netrc** file format.

---

rexecd Daemon

Purpose

Provides the server function for the **rexec** command.

Syntax

> **Note:** The **rexecd** daemon is normally started by the **inetd** daemon. It can also be controlled from the command line, using SRC commands.

/usr/sbin/rexecd [**-sc**]

Description

The /**usr**/**sbin**/**rexecd** daemon is the server for the **rexec** command. This daemon processes commands issued by a foreign host and returns the output of those commands to that foreign host. The **rexecd** daemon sends and receives data over a Transmission Control Protocol/Internet Protocol (TCP/IP) connection.

Changes to the **rexecd** daemon can be made using Web-based System Manager, the System Management Interface Tool (SMIT) or System Resource Controller (SRC), by editing the **/etc/inetd.conf** or /**etc**/**services** file. Entering `rexecd` at the command line is not recommended.The **rexecd** daemon is started by default when it is uncommented in the **/etc/inetd.conf** file.

The **inetd** daemon get its information from the /**etc**/**inetd.conf** file and the /**etc**/**services** file.

After changing the /**etc**/**inetd.conf** file, run the **refresh -s inetd** or **kill -1** *InetdPID* command to inform the **inetd** daemon of the changes to its configuration file.

Flags

-s Enables socket-level debugging.

-c Prevents reverse name resolution. When the **-c** flag is not specified, the **rexecd** daemon will fail if the reverse name resolution of the client fails.

Service Request Protocol

When the **rexecd** daemon receives a request, it initiates the following protocol:

1. The server reads characters from the socket up to a null (\0) byte and interprets the resulting string as an ASCII number (decimal).
2. If the number received is nonzero, the **rexecd** daemon interprets it as the port number of a secondary stream to be used for standard error output. The **rexecd** daemon then creates a second connection to the specified port on the client machine.
3. The **rexecd** daemon retrieves a null-terminated user name of up to 16 characters on the initial socket.

Related Information

The **kill** command, **lssrc** command, **refresh** command, **rexec** command, **startsrc** command, **stopsrc** command.

The **inetd** daemon.

The **rexec** subroutine.

The **/etc/inetd.conf** file format.

TCP/IP Overview for System Management in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.2 Web-based System Manager Administration Guide*.

TCP/IP Daemons in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

---

rgb Command

Purpose

Creates the database used by the X-Window system server for colors.

Syntax

rgb [*DatabaseName*] [*<InputfileName*]

Description

The **rgb** command reads lines from standard input and inserts them into its database to associate color names with specific red, green, and blue (RGB) values.

The **rgb** command produces two output files: *DatabaseName*.dir and *DatabaseName*.pag. If you do not specify a database file name, the default names **rgb.dir** and **rgb.pag** are used.

Each color entry is in the form:

```
Red Green Blue Colorname
```

where the *Red*, *Green*, and *Blue* elements are integer values ranging from 0-255. The actual color is determined by how the elements are combined. Each element can range from no intensity (0) to full intensity (255). The *Colorname* parameter can be descriptive or fanciful. For example, the sequence `250 250 250` could be named `white` or `snow`. Two or more entries can share the same element numbers or names.

Parameters

DatabaseName	Specifies the database to create for the output data.
<InputFileName	Specifies the name of the input file.

Examples

1. The following example shows a portion of an input file:

```
248 248 255     ghost white
245 245 245     white smoke
255 250 240     floral white
```

```
253 245 230      old lace
250 240 230      linen
255 218 185      peach puff
255 248 220      cornsilk
255 250 205      lemon chiffon
245 255 250      mint cream
240 255 255      azure
```

2. The following example generates the output files **Newcolor.dir** and **Newcolor.pag**.

```
rgb Newcolor < rgb.txt
```

where **Newcolor** is the *DatabaseName* and **rgb.txt** is the *InputFileName*.

Files

/usr/lib/X11/rgb.txt	The default rgb database input file.

---

ripquery Command

Purpose

Queries the RIP gateways.

Syntax

ripquery [**-1**] [**-2**] [**-**[**a5**] *authkey*] [**-n**] [**-N** *dest***[/***mask***]**] [**-p**] [**-r**] [**-v**] [**-w** *time*] *gateway*...

Description

The **ripquery** command is used to request all routes known by a RIP *gateway* by sending a RIP **REQUEST** or **POLL** command. The routing information in any routing packets returned is displayed numerically and symbolically. The **ripquery** command is intended to be used as a tool for debugging *gateway*s, not for network management. SNMP is the preferred protocol for network management.

Flags

-1	Send the query as a version 1 packet.
-2	Send the query as a version 2 packet (default).
-[**a5**] *authkey*	Specifies the authentication password to use for queries. If **-a** is specified, an authentication type of SIMPLE will be used, if **-5** is specified, an authentication type of MD5 will be used, otherwise the default is an authentication type of NONE. Authentication fields in incoming packets will be displayed, but not validated.
-n	Prevents the address of the responding host from being looked up to determine the symbolic name.
-N *dest***[/***mask***]**	Specifies that the query should be for the specified *dest/mask* instead of complete routing table. The specification of the optional mask implies a version 2 query. Up to 23 requests about specific destinations may be included in one packet.
-p	Uses the RIP **POLL** command to request information from the routing table. This is the default. If there is no response to the RIP **POLL** command, the RIP **REQUEST** command is tried. **gated** responds to a **POLL** command with all the routes learned via RIP.
-r	Uses the RIP **REQUEST** command to request information from the *gateway*'s routing table. Unlike the RIP **POLL** command, all *gateway*s should support the RIP **REQUEST**. If there is no response to the RIP **REQUEST** command, the RIP **POLL** command is tried. **gated** responds to a **REQUEST** command with all the routes he anounces out the specified interface.
-v	Version information about **ripquery** is displayed before querying the *gateway*s.

-w *time* Specifies the time in seconds to wait for the initial response from a *gateway*. The default value is 5 seconds.

Related Information

The **gated** Daemon.

---

rlogin Command

Purpose

Connects a local host with a remote host.

Syntax

rlogin *RemoteHost* [**-e** *Character*] [**-8**] [**-l** *User*] [**-f** | **-F**] [**-k** *realm*]

Description

The **/usr/bin/rlogin** command logs into a specified remote host and connects your local terminal to the remote host.

The remote terminal type is the same as that given in the **TERM** local environment variable. The terminal or window size is also the same, if the remote host supports them, and any changes in size are transferred. All echoing takes place at the remote host, so except for delays, the terminal connection is transparent. The Ctrl-S and Ctrl-Q key sequences stop and start the flow of information, and the input and output buffers are flushed on interrupts.

Remote Command Execution

When using the **rlogin** command, you can create a link to your path using a host name as the link name. For example:

```
ln -s /usr/bin/rsh HostName
```

Entering the host name specified by the *HostName* parameter with an argument (command) at the prompt, automatically uses the **rsh** command to remotely execute the command specified on the command line of the remote host specified by the *HostName* parameter.

Entering the host name specified by the *HostName* parameter without an argument (command) at the prompt, automatically uses the **rlogin** command to log in to the remote host specified by the *HostName* parameter.

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

The **rlogin** command execs (using the **exec** command) the **/usr/sbin/login** file to validate a user. This 1) allows all user and device attributes to take effect on telnet connections and 2) causes remote logins to count against the maximum number of login sessions allowable at a time (determined by the `maxlogins` attribute). Attributes are defined in the **/etc/security/user** and **/etc/security/login.cfg** files.

POSIX Line Discipline

The **rlogind** and **telnetd** daemons use POSIX line discipline to change the line discipline on the local TTY. If POSIX line discipline is not used on the local TTY, echoing other line disciplines may result in improper behavior. TCP/IP must have POSIX line discipline to function properly.

Flags

-8	Allows an 8-bit data path at all times. Otherwise, unless the start and stop characters on the remote host are not Ctrl-S and Ctrl-Q, the **rlogin** command uses a 7-bit data path and parity bits are stripped.
-e *Character*	Changes the escape character. Substitute the character you choose for *Character*.
-f	Causes the credentials to be forwarded. This flag will be ignored if Kerberos 5 is not the current authentication method. Authentication will fail if the current DCE credentials are not marked forwardable.
-F	Causes the credentials to be forwarded. In addition, the credentials on the remote system will be marked forwardable (allowing them to be passed to another remote system). This flag will be ignored if Kerberos 5 is not the current authentication method. Authentication will fail if the current DCE credentials are not marked forwardable.
-k *realm*	Allows the user to specify the realm of the remote station if it is different from the local systems realm. For these purposes, a realm is synonymous with a DCE cell. This flag will be ignored if Kerberos 5 is not the current authentication method.
-l *User*	Changes the remote user name to the one you specify. Otherwise, your local user name is used at the remote host.

Security

There are multiple authentication methods, each requiring different things to be set in order to allow the connection.

For Standard Authentication

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

- The local host is included in the remote **/etc/hosts.equiv** file, the local user is not the root user, and the **-l** *User* flag is not specified.
- The local host and user name is included in the **$HOME/.rhosts** file in the remote user account.

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

For Kerberos 5 Authentication

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

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

Unless otherwise modified by the **-e** flag, the standard escape character is a tilde (~). The escape character is only recognized by the remote host if it occurs at the beginning of a line. Otherwise, the escape character is sent to the remote host as a normal character. To send the escape character to the remote host as a normal character at the beginning of a line, press the escape character twice. Pressing the escape character and a period (for example, ~.) immediately disconnects the local terminal from the remote host.

Examples

1. To log in to a remote host with your local user name, enter:

   ```
   rlogin host2
   ```

   You are prompted to enter your password and then are logged in to the remote host `host2`.

To logoff the remote host, and close the connection, enter ~. (tilde, period).

2. To log in to a remote host with a different user name, enter:

   ```
   rlogin host2  -l dale
   ```

   You are prompted to enter your password and then are logged in to the remote host `host2` with the user name `dale`.

   To logoff the remote host, and close the connection, enter ~. (tilde, period).

3. To log in to a remote host with your local user name and change the escape character, enter:

   ```
   rlogin host2  -e\
   ```

   You are prompted to enter your password and then are logged in to the remote host `host2`. The escape character has been changed to \ (backslash).

   To logoff the remote host, and close the connection, enter \. (backslash, period).

Related Information

The **ftp** command, **login** command, **rcp** command, **rexec** command, **rsh** command, **telnet** command.

The **rlogind** and **krlogind** daemon.

The **kvalid_user** function.

The **hosts.equiv** file format, **.rhosts** file format.

Network Overview in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

Secure Rcmds in *AIX 5L Version 5.2 System User's Guide: Communications and Networks*.

---

rlogind Daemon

Purpose

Provides the server function for the **rlogin** command.

Syntax

Note: The **rlogind** daemon is normally started by the **inetd** daemon. It can also be controlled from the command line, using SRC commands.

/usr/sbin/rlogind [**-a**] [**-c**] [**-l**] [**-n**] [**-s**]

Description

The **/usr/sbin/rlogind** daemon is the server for the **rlogin** remote login command. The server provides a remote login facility.

Changes to the **rlogind** daemon can be made using Web-based System Manager, the System Management Interface Tool (SMIT) or System Resource Controller (SRC), by editing the **/etc/inetd.conf** or /**etc**/**services** file. Entering `rlogind` at the command line is not recommended. The **rlogind** daemon is started by default when it is uncommented in the **/etc/inetd.conf** file.

The **inetd** daemon get its information the /**etc**/**inetd.conf** file and the /**etc**/**services** file.

After changing the **/etc/inetd.conf** or /**etc**/**services** file, run the **refresh -s inetd** or **kill -1** *InetdPID* command to inform the **inetd** daemon of the changes to its configuration file.

Service Request Protocol

When the **rlogind** daemon receives a service request, the daemon initiates the following protocol:

1. The **rlogind** daemon checks the source port number for the request. If the port number is not in the range 512-1023, the **rlogind** daemon terminates the connection.
2. The **rlogind** daemon uses the source address of the initial connection request to determine the name of the client host. If the name cannot be determined, the **rlogind** daemon uses the dotted-decimal representation of the client host address.

Error Messages

The following error messages are associated with the **rlogind** daemon:

Try again.	A **fork** command made by the server has failed.
/usr/bin/shell:	No shell. The shell specified for the shell variable cannot be started. The shell variable may also be a program.

Flags

-a Disables **pty** speed enhancement feature.
-c Suppresses the reverse hostname lookup.
-l Prevents any authentication based on the user's **$HOME/.rhosts** file. However, a root user is automatically logged in when there is a **.rhosts** file in root's home directory as specified by the **/etc/passwd** file.
-n Disables transport-level keep-alive messages. The messages are enabled by default.
-s Turns on socket level debugging.

Examples

Note: The arguments for the **rlogind** daemon can be specified by using SMIT or by editing the **/etc/inetd.conf** file.

1. To start the **rlogind** daemon, enter the following:

   ```
   startsrc -t rlogin
   ```

   This command starts the **rlogind** subserver.
2. To stop the **rlogind** daemon normally, enter the following:

   ```
   stopsrc -t rlogin
   ```

   This command allows all pending connections to start and existing connections to complete but prevents new connections from starting.
3. To force stop the **rlogind** daemon and all **rlogind** connections, enter the following:

   ```
   stopsrc -t -f rlogin
   ```

   This command terminates all pending connections and existing connections immediately.
4. To display a short status report about the **rlogind** daemon, enter the following:

   ```
   lssrc -t rlogin
   ```

   This command returns the daemon's name, process ID, and state (active or inactive).

Related Information

The **kill** command, **lssrc** command, **refresh** command, **rlogin** command, **startsrc** command, **stopsrc** command.

The **rshd** daemon, **inetd** daemon, **syslogd** daemon.

The **pty** special file.

The **$HOME/.rhosts** file format, **/etc/hosts.equiv** file format, **/etc/inetd.conf** file format.

TCP/IP Daemons in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.2 Web-based System Manager Administration Guide*.

---

rm Command

Purpose

Removes (unlinks) files or directories.

Syntax

rm [**-f**] [**-r**] [**-R**] [**-i**] [**-e**] *File ...*

Description

The **rm** command removes the entries for the specified *File* parameter from a directory. If an entry is the last link to a file, the file is then deleted. If you do not have write permission for a file and the standard input is a terminal, you are prompted with the file name and ask to confirm that you want to delete the file. If you type a y (for yes), the file is deleted, type any other character and the file is not deleted. You do not need read or write permission for the file you want to remove. However, you must have write permission for the directory containing the file.

If the file is a symbolic link, the link is removed, but the file or directory that the symbolic link refers to remains. You do not need write permission to delete a symbolic link, if you have write permission in the directory.

If either of the files **.** (dot) or **..** (dot, dot) are specified as the base name portion of the *File* parameter, the **rm** command writes a diagnostic message to standard error and does nothing more with such parameters.

The **rm** command writes a prompt to standard error and reads a line from standard input if the **-f** flag is not specified, and either the *File* parameter does not have write permission and the standard input is a workstation, or the **-i** flag is specified. If the response is not affirmative, the **rm** command does nothing more with the current file and proceeds to the next file.

The files owned by other users cannot be removed if the sticky bit of the directory is set and the directory is not owned by the user.

> **Note:** The **rm** command supports the **—** (dash, dash) parameter as a delimiter that indicates the end of the flags.

Flags

-e Displays a message after each file is deleted.

-f Does not prompt before removing a write-protected file. Does not display an error message or return error status if a specified file does not exist. If both the **-f** and **-i** flags are specified, the last one specified takes affect.

-i Prompts you before deleting each file. When you use the **-i** and **-r** flags together, the **rm** command also prompts before deleting directories. If both the **-i** and **-f** flags are specified, the last one specified takes affect.

-r Permits recursive removal of directories and their contents when the *File* parameter is a directory. This flag is equivalent to the **-R** flag.

-R Permits recursive removal of directories and their contents when the *File* parameter is a directory. This flag is equivalent to the **-r** flag.

Exit Status

This command returns the following exit values:

0 If the **-f** flag was not specified, all the named directory entries were removed; otherwise, all the existing named directory entries were removed.

>0 An error occurred.

Examples

1. To delete a file, enter:

   ```
   rm myfile
   ```

   If there is another link to this file, then the file remains under that name, but the name `myfile` is removed. If `myfile` is the only link, the file itself is deleted.

2. To delete a file without first receiving a confirmation prompt, enter:

   ```
   rm  -f core
   ```

   No confirmation prompt is issued before the **rm -f** command attempts to remove the file named `core`. However, an error message displays if the `core` file is write-protected and you are not the owner of the file or you do not have root authority. No error message displays when the **rm -f** command attempts to remove nonexistent files.

3. To delete files one by one, enter:

   ```
   rm  -i mydir/*
   ```

   After each file name is displayed, enter `y` to delete the file, or press the Enter key to keep it.

4. To delete a directory tree, enter:

   ```
   rm -i r manual
   ```

   This command recursively removes the contents of all subdirectories of the `manual` directory, prompting you regarding the removal of each file, and then removes the `manual` directory itself, for example:

   ```
   You:  rm -ir manual
   System: rm: Select files in directory manual? Enter y for yes.
   You:  y
   System: rm: Select files in directory manual/draft1? Enter y for yes.
   You:  y
   System: rm: Remove manual/draft1?
   You:  y
   System: rm: Remove manual/draft1/chapter1?
   You:  y
   System: rm: Remove manual/draft1/chapter2?
   You:  y
   ```

```
System: rm: Select files in directory manual/draft2? Enter y for yes.
You:  y
System: rm: Remove manual/draft2?
You:  y
System: rm: Remove manual?
You:  y
```

Here, the **rm** command first asks if you want it to search the `manual` directory. Because the `manual` directory contains directories, the **rm** command next asks for permission to search `manual/draft1` for files to delete, and then asks if you want it to delete the `manual/draft1/chapter1` and `manual/draft1/chapter2` files. The **rm** command next asks for permission to search the `manual/draft2` directory. Then asks for permission to delete the `manual/draft1`, `manual/draft2`, and `manual` directories.

If you deny permission to remove a subdirectory (for example, `manual/draft2`), the **rm** command does not remove the `manual` directory. Instead, you see the message: `rm: Directory manual not empty`.

Files

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

Related Information

The **ln** command.

The **unlink** subroutine.

Directory Overview in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices*.

File and Directory Access Modes in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*.

Linking Files and Directories in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices*.

---

rm_niscachemgr Command

Purpose

Stops the **nis_cachemgr** daemon and comments the entry in the **/etc/rc.nfs** file.

Syntax

rm_niscachemgr [**-I** | **-B** | **-N**]

Description

The **rm_niscachemgr** command comments the entry for the **nis_cachemgr** daemon in the **/etc/rc.nfs** file. The **rm_niscachemgr** daemon stops the **nis_cachemgr** daemon by using the **stopsrc** command.

Note: The **mk_nisd**, **mk_cachemgr**, **mk_nispasswdd**, **rm_nisd**, **rm_cachemgr**, and **rm_nispasswdd** commands do two things:

- Alter the entries of daemon startup calls in **/etc/rc.nfs**.
- Alter the default behavior of the daemon **src** entities.

For example, if the **rpc.nisd** daemon is supposed to start with the **-Y** flag, this will not be explicitly set in the **/etc/rc.nfs** entry for starting the **rpc.nisd** daemon. Instead, a **chssys** is executed to place the default options which are added (if any) to the daemons during startup. To verify that these options exist, use the **lssrc -S -s** *subsystem* command to show the default options.

Flags

-I Comments the entry for the **nis_cachemgr** daemon in the **/etc/rc.nfs** file.
-B Comments the entry for the **nis_cachemgr** daemon in the **/etc/rc.nfs** file and stops the **nis_cachemgr** daemon. This flag is the default.
-N Stops the **nis_cachemgr** daemon using the **stopsrc** command. This flag does not change the **/etc/rc.nfs** file.

Note: An important effect of running this command is that the domain name of the NIS+ environment will be reset to NULL. It is assumed that if the administrator turns off the **nis_cachemgr**, the NIS+ configuration is no longer active. By resetting the domain name, unnecessary lookups are prevented. If the administrator does not desire this effect, they should run **chypdom** after **rm_niscachemgr** has been run.

Examples

To comment the entry in the **/etc/rc.nfs** file that starts the **nis_cachemgr** daemon, enter:

```
rm_niscachemgr -I
```

This command will not stop the currently executing daemon.

Files

/etc/rc.nfs Contains the startup script for the NFS and NIS daemons.

Related Information

The **smit** command

The **nis_cachemgr** daemon.

Network Information Service (NIS+) in *AIX 5L Version 5.2 Network Information Services (NIS and NIS+) Guide*

Network File System (NFS) Overview for System Management in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*

How to Start the NFS Daemons, How to Stop the NFS Daemons, Exporting a File System Using Secure NFS, Mounting a File System Using Secure NFS in *AIX 5L Version 5.2 Network Information Services (NIS and NIS+) Guide*

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.2 Web-based System Manager Administration Guide*.

System Management Interface Tool in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices*.

NIS+ Reference.

---

rm_nisd Daemon

Purpose

Stops the **rpc.nisd** daemon and comments the entry in the **/etc/rc.nfs** file.

Syntax

rm_nisd [**-I** | **-B** | **-N**]

Description

The **rm_nisd** daemon comments the entry for the **rpc.nisd** daemon in the **/etc/rc.nfs** file. The **rm_nisd** daemon stops the **rpc.nisd** daemon by using the **stopsrc** command.

> **Note:** The **mk_nisd**, **mk_cachemgr**, **mk_nispasswdd**, **rm_nisd**, **rm_cachemgr**, and **rm_nispasswdd** commands do two things:
>
> - Alter the entries of daemon startup calls in **/etc/rc.nfs**.
> - Alter the default behavior of the daemon **src** entities.
>
> For example, if the **rpc.nisd** daemon is supposed to start with the **-Y** flag, this will not be explicitly set in the **/etc/rc.nfs** entry for starting the **rpc.nisd** daemon. Instead, a **chssys** is executed to place the default options which are added (if any) to the daemons during startup. To verify that these options exist, use the **lssrc -S -s** *subsystem* command to show the default options.

Flags

-I Comments the entry for the **rpc.nisd** daemon in the **/etc/rc.nfs** file.

-B Comments the entry for the **rpc.nisd** daemon in the **/etc/rc.nfs** file and stops the **rpc.nisd** daemon. This flag is the default.

-N Stops the **rpc.nisd** daemon using the **stopsrc** command. This flag does not change the **/etc/rc.nfs** file.

Examples

To comment the entry in the **/etc/rc.nfs** file that starts the **rpc.nisd** daemon, enter:

```
rm_nisd -I
```

This command will not stop the currently executing daemon.

Files

/etc/rc.nfs Contains the startup script for the NFS and NIS daemons.

Related Information

The **smit** command.

The **rpc.nisd** daemon.

Network Information Service (NIS+) Overview for System Management in *AIX 5L Version 5.2 Network Information Services (NIS and NIS+) Guide*

Network File System (NFS) Overview for System Management in *AIX 5L Version 5.2 Network Information Services (NIS and NIS+) Guide*

How to Start the NFS Daemons, How to Stop the NFS Daemons, in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*

Exporting a File System Using Secure NFS and Mounting a File System Using Secure NFS in *AIX 5L Version 5.2 Security Guide*.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.2 Web-based System Manager Administration Guide*.

System Management Interface Tool in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices*.

NIS+ Reference.

---

rm_nispasswdd Daemon

Purpose

Stops the **rpc.nispasswdd** daemon and comments the entry in the **/etc/rc.nfs** file.

Syntax

rm_nispasswdd [**-I** | **-B** | **-N**]

Description

The **rm_nispasswdd** daemon comments the entry for the **rpc.nispasswdd** daemon in the **/etc/rc.nfs** file. The **rm_nispasswdd** daemon stops the **rpc.nispasswdd** daemon by using the **stopsrc** command.

> **Note:** The **mk_nisd**, **mk_cachemgr**, **mk_nispasswdd**, **rm_nisd**, **rm_cachemgr**, and **rm_nispasswdd** commands do two things:
>
> - Alter the entries of daemon startup calls in **/etc/rc.nfs**.
> - Alter the default behavior of the daemon **src** entities.
>
> For example, if the **rpc.nisd** daemon is supposed to start with the **-Y** flag, this will not be explicitly set in the **/etc/rc.nfs** entry for starting the **rpc.nisd** daemon. Instead, a **chssys** is executed to place the default options which are added (if any) to the daemons during startup. To verify that these options exist, use the **lssrc -S -s** *subsystem* command to show the default options.

Flags

-I Comments the entry for the **rpc.nispasswdd** daemon in the **/etc/rc.nfs** file.

-B Comments the entry for the **rpc.nispasswdd** daemon in the **/etc/rc.nfs** file and stops the **rpc.nispasswdd** daemon. This flag is the default.

-N Stops the **rpc.nispasswdd** daemon using the **stopsrc** command. This flag does not change the **/etc/rc.nfs** file.

Examples

To comment the entry in the **/etc/rc.nfs** file that starts the **rpc.nispasswdd** daemon, enter:

```
rm_nispasswdd -I
```

This command will not stop the currently executing daemon.

Files

/etc/rc.nfs Contains the startup script for the NFS and NIS daemons.

Related Information

The **smit** command.

The **rpc.nispasswdd** daemon.

Network Information Service (NIS+) Overview for System Management in *AIX 5L Version 5.2 Network Information Services (NIS and NIS+) Guide*

Network File System (NFS) Overview for System Management in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*

How to Start the NFS Daemons, How to Stop the NFS Daemons, in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*

Exporting a File System Using Secure NFS and Mounting a File System Using Secure NFS in *AIX 5L Version 5.2 Security Guide*.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.2 Web-based System Manager Administration Guide*.

NIS+ Reference.

---

rmail Command

Purpose

Handles remote mail received through Basic Networking Utilities (BNU).

Syntax

rmail *User*

Description

The **rmail** command interprets incoming mail received through the **uucp** command. It collapses `From` header lines in the form generated by the **bellmail** command into a single line of the form:

```
return-path!sender
```

The **rmail** command passes the processed mail on to the **sendmail** command. The *User* parameter must specify a user recognized by the **sendmail** command.

Related Information

The **bellmail** command, **sendmail** command, **uucp** command.

Mail Overview for System Management in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

---

rmaudrec Command

Purpose

Removes records from the audit log.

Syntax

rmaudrec [**–a** | **–n** *node_name1*[**,***node_name2*]...] [**–S** *subsystem_name*]
–s *selection_string* [**–h**] [**–V**]

Description

The **rmaudrec** command is used to delete records in the audit log.

The audit log is a facility for recording information about the system's operation. It can include information about the normal operation of the system as well as failures and other errors. It is meant to augment the error log functionality by conveying the relationship of the error relative to other system activities. All detailed information about failures is still written to the AIX error log.

Records are created in the audit log by subsystems that have been instrumented to do that. For example, the event response subsystem runs in the background to monitor administrator-defined conditions and then invokes one or more actions when a condition becomes true. Because this subsystem runs in the background, it is difficult for the operator or administrator to understand the total set of events that occurred and the results of any actions that were taken in response to an event. Because the event response subsystem records its activity in the audit log, the administrator can easily view its activity as well as that of other subsystems. In addition, records may sometimes need to be removed explicitly, which can be done using this command.

Each record in the audit log contains named fields. Each field contains a value that provides information about the situation corresponding to the record. For example, the field named **Time** indicates the time at which the situation occurred. Each record has a set of common fields and a set of subsystem-specific fields. The common fields are present in every record in the audit log. The subsystem-specific fields vary from record to record. Their names are only significant when used with a subsystem name because they may not be unique across all subsystems. Each record is derived from a template that defines which subsystem-specific fields are present in the record and defines a format string that is used to generate a message describing the situation. The format string may use record fields as inserts. A subsystem typically has many templates.

The field names can be used as variables in a *selection string* to choose which records are deleted. The selection string is matched against each record using the referenced fields of each record to perform the match. Any records that match will be removed. The selection string is specified with the **-s** flag.

A selection string is an expression composed of field names, constants and operators. The syntax of a selection string is very similar to an expression in the C programming language. For information on how to specify selection strings, see *RSCT Guide and Reference*.

The common field names are:

Time Specifies the time when the situation occurred that the record corresponds to. The value is a 64-bit integer and represents the number of microseconds since Unix Epoch (00:00:00 GMT January 1, 1970). See the constants below for specifying the time in more user-friendly formats.

Subsystem
Specifies the subsystem that generated the record. This is a string.

Category
Indicates the importance of the situation corresponding to the audit record, as determined by the subsystem that generated the record. The valid values are: **0** (informational) and **1** (error).

SequenceNumber
Specifies the unique 64-bit integer that is assigned to the record. No other record in the audit log will have the same sequence number.

TemplateId
Specifies the subsystem-dependent identifier that is assigned to records that have the same content and format string. This value is a 32-bit unsigned integer.

NodeName
Specifies the name of the node from which the record was obtained. This field name cannot be used in a selection string.

In addition to the constants in expressions that are described in *RSCT Guide and Reference*, you can use the following syntax for dates and times with this command:

#*mmddhhmmyyyy*
This format consists of a sequence of decimal characters that are interpreted according to the pattern shown. The fields in the pattern are, from left to right: *mm* = month, *dd* = day, *hh* = hour, *mm* = minutes, *yyyy* = year. For example, **#010523042002** corresponds to January 5, 11:04 PM, 2002. The fields can be omitted from right to left. If not present, the following defaults are used: year = the current year, minutes = 0, hour = 0, day = 1, and month = the current month.

#-mmddhhmmyyyy
This format is similar to the previous one, but is relative to the current time and date. For example, the value **#-0001** corresponds to one day ago and the value **#-010001** corresponds to one month and one hour ago. Fields can be omitted starting from the right and are replaced by 0.

The audit records considered for deletion and matched against the selection string can be restricted to a specific subsystem by using the **-S** flag. If this flag is specified, the subsystem-specific field names can be used in the selection string in addition to the common field names.

The nodes from which audit log records are considered for deletion can be restricted to a set of specific nodes by using the **-n** flag. If this flag is specified, the search will be limited to the set of nodes listed. Otherwise, the search will be performed for all nodes defined within the current management scope as determined by the CT_MANAGEMENT_SCOPE environment variable.

It is advisable to first use the **lsaudrec** command with the same **-s** and **-n** flag values to list the records that will be deleted. This minimizes the possibility of the selection string matching more records than intended.

Flags

-a Specifies that records from all nodes in the domain are to be removed. If both the **-n** and the **-a** flags are omitted, records from the local node only are removed.

-n *node_name1***[,***node_name2***]...**
Specifies the list of nodes containing audit log records that will be examined and considered for deletion if they meet the other criteria, such as matching the specified selection string. Node group names can also be specified, which are expanded into a list of node names. If both the **-n** and the **-a** flags are omitted, records from the local node only will be deleted.

-S *subsystem_name*
Specifies a subsystem name. If this flag is present, only records identified by *subsystem_name* are considered for deletion. The records to be deleted can be further restricted by the **-s** flag. If the subsystem name contains any spaces, it must be enclosed in single or double quotation marks.

For backward compatibility, the subsystem name can be specified using the **-n** flag *only* if the **-a** and the **-S** flags are *not* specified.

-s *selection string*
Specifies a selection string. This string is evaluated against each record in the audit log. If the evaluation results in a non-zero result (**TRUE**), the record is removed from the audit log. If the selection string contains any spaces, it must be enclosed within single or double quotation marks. For information on how to specify selection strings, see *RSCT Guide and Reference*.

The names of fields within the record can be used in the expression. If the **-S** flag is not specified, only the names of common fields can be used. See the **Description** for a list of the common field names and their data types. If the **-S** flag is specified, the name of any field for the specified subsystem as well as the common field names can be used.

If this flag is not specified, no records will be removed from the audit log.

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

-V Writes the command's verbose messages to standard error.

Parameters

field_name1 **[***field_name2***...]**
Specifies one or more fields in the audit log records to be displayed. The order of the field names on the command line corresponds to the order in which they are displayed. If no field names are specified, **Time**, **Subsystem**, **Severity**, and **Message** are displayed by default. If the management scope is not local, **NodeName** is displayed as the first column by default. See the **Description** for information about these and other fields.

Security

In order to remove records from an audit log when the **-S** flag is omitted, a user must have write access to the target resource class on each node from which records are to be removed. When the **-S** flag is specified, the user must have write access to the audit log resource corresponding to the subsystem identified by the **-S** flag on each node from which records are to be removed.

Authorization is controlled by the RMC access control list (ACL) file that exists on each node.

Exit Status

0 The command ran successfully.

1 An error occurred with RMC.

2 An error occurred with a command-line interface script.

3 An incorrect flag was entered on the command line.

4 An incorrect parameter was entered on the command line.

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

Environment Variables

CT_CONTACT
Determines the system where the session with the resource monitoring and control (RMC) daemon is established. When CT_CONTACT is set to a host name or IP address, the command contacts the RMC daemon on the specified host. If CT_CONTACT is not set, the command contacts the RMC daemon on the local system where the command is being run. The target of the RMC daemon session and the management scope determine the resource classes or resources that can be affected by this command.

CT_MANAGEMENT_SCOPE
Determines (in conjunction with the **-a** and **-n** flags) the management scope that is used for the session with the RMC daemon. The management scope determines the set of possible target nodes where audit log records can be deleted. If the **-a** and **-n** flags are not specified, local scope is used. When either of these flags is specified, CT_MANAGEMENT_SCOPE is used to determine the management scope directly. The valid values are:

0 Specifies *local* scope.

1 Specifies *local* scope.

2 Specifies *peer domain* scope.

3 Specifies *management domain* scope.

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

Standard Output

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

Standard Error

If the **-V** flag is specified and the command completes successfully, a message indicating the number of records that were deleted will be written to standard error.

Examples

1. To remove all records from the audit log on every node in the management scope defined by the CT_MANAGEMENT_SCOPE environment variable, enter:

   ```
   rmaudrec -s "Time > 0"
   ```

   or

   ```
   rmaudrec -s "SequenceNumber >= 0"
   ```

2. To remove all records more than a week old on every node in the management scope defined by the CT_MANAGEMENT_SCOPE environment variable, enter:

   ```
   rmaudrec -s "Time < #-0007"
   ```

3. To remove all records that are more than a day old and created by the **abc** subsystem on nodes **mynode** and **yournode**, enter:

   ```
   rmaudrec -S abc -s "Time < #-0001" -n mynode,yournode
   ```

Location

/usr/sbin/rsct/bin/rmaudrec Contains the **rmaudrec** command

Related Information

Commands: **lsaudrec**

---

rmC2admin Command

Purpose

Remove the configuration files for a distributed C2 System host.

Syntax

rmC2admin [**-m**]

Description

The **rmC2admin** command replaces the distributed C2 System symbolic links with the actual files. The directory **/etc/data.shared** will be removed. When the **-m** flag is used, the **hd10sec** file system and **/etc/data.master** directory will be removed as well. This option should only be used after all other hosts in the C2 System have replaced their administrative host with another system or removed the C2 configuration files as well.

The entries for the system initialization scripts in **/etc/inittab** will also be removed, and rebooting this system will result in the system not being configured for C2 mode.

Executing this command in multi-user mode will result in the user definitions from the C2 System being retained. Executing this command in single-user mode will result in the user definitions from the C2 System being removed and the root user being the only valid user ID.

The system should be rebooted immediately after executing this command so that the changes may take effect.

Flags

-m	The host was configured as the administrative master

Exit Status

0	The C2 System administrative host information has been successfully removed.
1	The system was not configured to operate in C2 mode.
2	The system was not installed with the C2 option.
3	An error occurred removing the C2 System administrative host information.
4	An invalid command line option was used.

Files

/usr/sbin/rmC2admin	Contains the **rmC2admin** command.

Related Information

The **chC2admin** command, **isC2host** command, **lsC2admin** command, **mkC2admin** command.

---

rmCCadmin Command

Purpose

Remove the configuration files for a distributed Common Criteria enabled System host.

Syntax

rmCCadmin [**-m**]

Description

The **rmCCadmin** command replaces the distributed Common Criteria enabled System symbolic links with the actual files. The directory **/etc/data.shared** will be removed. When the **-m** flag is used, the **hd10sec** file system and **/etc/data.master** directory will be removed as well. This option should only be used after all other hosts in the Common Criteria enabled System have replaced their administrative host with another system or removed the Common Criteria enabled configuration files as well.

The entries for the system initialization scripts in **/etc/inittab** will also be removed, and rebooting this system will result in the system not being configured for Common Criteria enabled mode.

Executing this command in multi-user mode will result in the user definitions from the Common Criteria enabled System being retained. Executing this command in single-user mode will result in the user definitions from the Common Criteria enabled System being removed and the root user being the only valid user ID.

The system should be rebooted immediately after executing this command so that the changes may take effect.

Flags

-m The host was configured as the administrative master

Exit Status

0 The Common Criteria enabled System administrative host information has been successfully removed.

1 The system was not configured to operate in Common Criteria enabled mode.

2 The system was not installed with the Common Criteria enabled option.

3 An error occurred removing the Common Criteria enabled System administrative host information.

4 An invalid command line option was used.

Files

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

Related Information

The **chCCadmin** command, **isCChost** command, **lsCCadmin** command, **mkCCadmin** command.

---

rmcctrl Command

Purpose

Manages the resource monitoring and control (RMC) subsystem.

Syntax

rmcctrl { **-a** | **-A** | **-d** | **-k** | **-K** | **-p** | **-P** | **-q** | **-Q** | **-s** | **-z** | **-h** }

Description

The **rmcctrl** command controls the operation of the resource monitoring and control (RMC) subsystem. The subsystem is under the control of the system resource controller (SRC) with a subsystem name of **ctrmc** and a subsystem group name of **rsct**. The RMC subsystem definition is added to the subsystem object class and then started when Reliable Scalable Cluster Technology (RSCT) is installed. In addition, an entry is made in the **/etc/inittab** file so that the RMC subsystem is started automatically when the system is booted up.

Note: While the RMC subsystem can be stopped and started by using the **stopsrc** and **startsrc** commands, it is recommended that the **rmcctrl** command be used to perform these functions.

Flags

–a Adds the RMC subsystem to the subsystem object class and places an entry at the end of the **/etc/inittab** file

–A Adds and starts the RMC subsystem.

–d Deletes the RMC subsystem from the subsystem object class and removes the RMC entry from the **/etc/inittab** file

–k Stops the RMC subsystem

–K Stops the RMC subsystem and all resource managers

–p Enables remote client connections

–P Disables remote client connections

–q Enables remote client connections the next time the RMC subsystem is started

–Q Disables remote client connections the next time the RMC subsystem is started

–s Starts the RMC subsystem

–z Stops the RMC subsystem and all resource managers, but the command does not return until the RMC subsystem and the resource managers are actually stopped

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

Security

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

Exit Status

0 Command has run successfully.

1 Command was not successful.

Restrictions

This command applies to the **cthags** and **cthats** subsystems only.

Standard Output

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

Standard Error

Error messages are written to standard error (and to the **ctsnap.***host_name***.***nnnnnnnn***.log** file).

Examples

1. To add the RMC subsystem, enter:
   ```
   rmcctrl -a
   ```
2. To start the RMC subsystem, enter:
   ```
   rmcctrl -s
   ```
3. To stop the RMC subsystem, enter:
   ```
   rmcctrl -k
   ```
4. To delete the RMC subsystem, enter:
   ```
   rmcctrl -d
   ```

Location

/usr/sbin/rsct/bin/rmcctrl Contains the **rmcctrl** 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**

---

rmclass Command

Purpose

Remove a Workload Management class.

Syntax

rmclass [**-d** *Config_Dir*] [**-S** *SuperClass*] *Name*

Description

The **rmclass** command removes the superclass or the subclass identified by the *Name* parameter from the class definition file, the class limits file and the class shares file. The class must already exist. The predefined **Default**, **System**, and **Shared** classes cannot be removed.

In addition, when removing a superclass **Super**, the directory **/etc/wlm/***Config_Dir***/Super** and all the WLM property files it contains (if they exist) are removed. Removing a superclass fails if any user created subclass still exists (subclass other than **Default** and **Shared**).

> **Note:** Only root can remove a superclass. Only root or authorized users whose user ID or group ID matches the user name or group name specified in the attributes **adminuser** and **admingroup** of a superclass can remove a subclass of this superclass.

Normally, **rmclass** deletes the class and its attributes in the relevant WLM property files, and the modifications are applied to the in-core class definitions (active classes) only after an update of WLM using the **wlmcntrl** command.

If an empty string is passed as the configuration name (*Config_dir*) with the **-d** flag, the class is deleted only in the WLM in-core data structures, and no property file is updated. So, if the class is still defined in a WLM configuration, it is recreated after an update or restart of WLM. This flag should be mainly used to remove classes dynamically created in the in-core WLM data structures only by applications using the WLM API, for example, to do some cleanup after application failure.

Note: This command cannot apply to a set of time-based configurations (do not specify a set with the **-d** flag). If the current configuration is a set, the **-d** flag must be given to indicate which regular configuration the command should apply to.

Flags

-d *Config_Dir* Uses **/etc/wlm/***Config_dir* as alternate directory for the properties files. If this flag is not used, the configuration files in the directory pointed to by **/etc/wlm/current** are used. If an empty string is passed as the configuration name (`-d ""`) the class is deleted only in the WLM in-core data structures and no configuration file is modified.

-S *SuperClass* Specifies the name of the superclass when removing a subclass. There are two ways of specifying the subclass **Sub** of superclass **Super**:

1. Specify the full name of the subclass as **Super.Sub** and do not use **-S**.
2. Specify the **-S** flag to give the superclass name and use the short name for the subclass:

   ```
   rmclass options -S Super  Sub
   ```

Files

classes Contains the names and definitions of the classes.
limits Contains the resource limits.
shares Contains the resource shares.

Related Information

The **wlmcntrl** command, **lsclass** command, **chclass** command, and **mkclass** command.

---

rmcomg Command

Purpose

Removes a communication group that has already been defined from a peer domain.

Syntax

rmcomg [**–q**] [**–h**] [**–TV**] *communication_group*

Description

The **rmcomg** command removes the definition of the existing communication group with the name specified by the *communication_group* parameter for the online peer domain. The communication group is used to define heartbeat rings for use by topology services and to define the tunables for each heartbeat ring. The communication group determines which devices are used for heartbeating in the peer domain.

The **rmcomg** command must be run on a node that is currently online in the peer domain where the communication group is defined. The communication group must not be referred to by an interface resource. Use the **chcomg** command to remove references by interface resources to a communication group.

Flags

-q Specifies quiet mode. The command does not return an error if the communication group does not exist.

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

-T Writes the command's trace messages to standard error. For your software service organization's use only.

-V Writes the command's verbose messages to standard output.

Parameters

communication_group
Specifies the name of the defined communication group that is to be removed from the peer domain.

Security

The user of the **rmcomg** command needs write permission for the **IBM.CommunicationGroup** resource class. By default, **root** on any node in the peer domain has read and write access to this resource class through the configuration resource manager.

Exit Status

0 The command ran successfully.

1 An error occurred with RMC.

2 An error occurred with a command-line interface script.

3 An incorrect flag was entered on the command line.

4 An incorrect parameter was entered on the command line.

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

6 The communication group does not exist.

Environment Variables

CT_CONTACT
Determines the system where the session with the resource monitoring and control (RMC) daemon occurs. When CT_CONTACT is set to a host name or IP address, the command contacts the RMC daemon on the specified host. If CT_CONTACT is not set, the command contacts the RMC daemon on the local system where the command is being run. The target of the RMC daemon session and the management scope determine the resource classes or resources that are processed.

Restrictions

This command must be run on a node that is defined and online to the peer domain where the communication group is to be removed.

Standard Input

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

Standard Output

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

Standard Error

All trace messages are written to standard error.

Examples

In this example, **nodeA** is defined and online to **ApplDomain**. To remove the communication group definition **ComGrp1** for the peer domain **ApplDomain**, run this command on **nodeA**:

```
rmcomg ComGrp1
```

Location

/usr/sbin/rsct/bin/rmcomg Contains the **rmcomg** command

Files

The access control list (ACL) file **/var/ct/cfg/ctrmc.acls** is modified. If this file does not exist, it is created.

Related Information

Commands: **chcomg**, **lscomg**, **lsrpdomain**, **lsrpnode**, **mkcomg**, **preprpnode**

---

rmcondition Command

Purpose

Removes a condition.

Syntax

rmcondition [**–f**] [**–q**] [**–h**] [**–TV**] *condition*[**:***node_name*]

Description

The **rmcondition** command removes the condition specified by the *condition* parameter. The condition must already exist to be removed. When the condition must be removed even if it has linked responses, use the **-f** flag to force the condition and the links with the responses to be removed. If the **-f** flag is not specified and links with responses exist, the condition is not removed. This command does not remove responses.

Flags

–f Forces the condition to be removed even if it is linked to responses. The links with the responses are removed as well as the condition, but the responses are not removed.

–q Does not return an error when *condition* does not exist.

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

–T Writes the command's trace messages to standard error. For your software service organization's use only.

–V Writes the command's verbose messages to standard output.

Parameters

condition Specifies the name of a condition to be removed.

node_name Specifies the node where the condition is defined. If *node_name* is not specified, the local node is used. *node_name* is a node within the scope determined by the CT_MANAGEMENT_SCOPE environment variable.

Security

The user needs write permission for the IBM.Condition resource class to run **rmcondition**. Permissions are specified in the access control list (ACL) file on the contacted system. See the *RSCT Guide and Reference* 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_MANAGEMENT_SCOPE

Determines the management scope that is used for the session with the RMC daemon in processing the resources of the event response resource manager (ERRM). The management scope determines the set of possible target nodes where the resources can be processed. The valid values are:

0 Specifies *local* scope.

1 Specifies *local* scope.

2 Specifies *peer domain* scope.

3 Specifies *management domain* scope.

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

Standard Output

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

Standard Error

All trace messages are written to standard error.

Examples

These examples apply to standalone systems:

1. To remove the condition definition named ″FileSystem space used″, run this command:

   ```
   rmcondition "FileSystem space used"
   ```

2. To remove the condition definition named ″FileSystem space used″ even if the condition is linked with responses, run this command:

   ```
   rmcondition -f "FileSystem space used"
   ```

This example applies to management domains:

1. In this example, the current node is the management server. To remove the condition definition named ″nodeB FileSystem space used″ that is defined on managed node **nodeB**, run this command:

   ```
   rmcondition "FileSystem space used:nodeB"
   ```

This example applies to peer domains:

1. To remove the condition definition named ″nodeA FileSystem space used″ that is defined on node **nodeA**, run this command from any node in the domain:

```
rmcondition "nodeA FileSystem space used:nodeA"
```

Location

/usr/sbin/rsct/bin/rmcondition
Contains the **rmcondition** command

Related Information

ERRM commands: **chcondition**, **lscondition**, **lscondresp**, **mkcondition**

Files: **rmccli**

Books: see the *RSCT Guide and Reference* for more information about ERRM operations

---

rmcondresp Command

Purpose

Deletes the link between a condition and one or more responses.

Syntax

To delete the link between a condition and one or more responses:

rmcondresp [**–q**] [**–h**] [**–TV**] *condition*[**:***node_name*] [*response* [*response*...]]

To delete all of the links to one or more responses:

rmcondresp [**–q**] **–r** [**–h**] [**–TV**] *response1* [*response2*...][**:***node_name*]

Description

The **rmcondresp** command deletes the link between a condition and one or more responses. The response is no longer run when the condition occurs. Use the **-r** flag to specify that the command parameters consist only of responses. This deletes all links to conditions for these responses. If only a condition is specified, links to all responses for that condition are deleted.

Flags

–q Does not return an error when either *condition* or *response* does not exist.

–r Indicates that all command parameters are responses. There are no conditions specified. This command removes links from all conditions that are linked to the specified responses.

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

–T Writes the command's trace messages to standard error. For your software service organization's use only.

–V Writes the command's verbose messages to standard output.

Parameters

condition Specifies the name of the condition linked to the response. The condition is always specified first unless the **-r** flag is used.

response Specifies the name of a response or more than one response. The links from the specified responses to the specified condition are removed.

node_name Specifies the node where the condition is defined. If the **-r** flag is used, it is the node where the response is defined. *node_name* is a node within the scope determined by the CT_MANAGEMENT_SCOPE environment variable.

Security

The user needs write permission for the IBM.Association resource class to run **rmcondresp**. Permissions are specified in the access control list (ACL) file on the contacted system. See the *RSCT Guide and Reference* 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_MANAGEMENT_SCOPE

Determines the management scope that is used for the session with the RMC daemon in processing the resources of the event response resource manager (ERRM). The management scope determines the set of possible target nodes where the resources can be processed. The valid values are:

0 Specifies *local* scope.

1 Specifies *local* scope.

2 Specifies *peer domain* scope.

3 Specifies *management domain* scope.

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

Standard Output

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

Standard Error

All trace messages are written to standard error.

Examples

These examples apply to standalone systems:

1. To delete the link between the condition ″FileSystem space used″ and the response ″Broadcast event on-shift″, run this command:

   ```
   rmcondresp "FileSystem space used" "Broadcast event on-shift"
   ```

2. To delete the links between the condition ″FileSystem space used″ and all of its responses, run this command:

   ```
   rmcondresp "FileSystem space used"
   ```

3. To delete the links between the condition ″FileSystem space used″ and the responses ″Broadcast event on-shift″ and ″E-mail root any time″, run this command:

   ```
   rmcondresp "FileSystem space used" "Broadcast event on-shift" "E-mail root any time"
   ```

4. To delete the links between the response ″Broadcast event on-shift″ and all of the conditions that use it, run this command:

   ```
   rmcondresp -r "Broadcast event on-shift"
   ```

These examples apply to management domains:

1. To delete the link between the condition ″FileSystem space used″ on the management server and the response ″Broadcast event on-shift″, run this command on the management server:

   ```
   rmcondresp "FileSystem space used" "Broadcast event on-shift"
   ```

2. To delete the links between the condition ″FileSystem space used″ on the managed node **nodeB** and the responses ″Broadcast event on-shift″ and ″E-mail root any time″, run this command on the management server:

   ```
   rmcondresp  "FileSystem space used":nodeB \
   "Broadcast event on-shift" "E-mail root any time"
   ```

These examples apply to peer domains:

1. To delete the links between the condition ″FileSystem space used″ on **nodeA** in the domain and the responses ″Broadcast event on-shift″ and ″E-mail root any time″, run this command on any node in the domain:

   ```
   rmcondresp "FileSystem space used":nodeA \
   "Broadcast event on-shift" "E-mail root any time"
   ```

2. To delete the links between all conditions on **nodeA** in the domain and the response ″Broadcast event on-shift″, run this command on any node in the domain:

   ```
   rmcondresp -r "Broadcast event on-shift":nodeA
   ```

Location

/usr/sbin/rsct/bin/rmcondresp
Contains the **rmcondresp** command

Related Information

ERRM commands: **lscondresp**, **mkcondition**, **mkcondresp**, **mkresponse**, **startcondresp**, **stopcondresp**

Files: **rmccli**

Books: see the *RSCT Guide and Reference* for more information about ERRM operations

rmdel Command

Purpose

Removes a delta from a SCCS file.

Syntax

rmdel -r *SID File ...*

Description

The **rmdel** command removes the delta specified by the *SID* variable from each Source Code Control System (SCCS) file indicated in the *File* parameter. You can remove only the most recently created delta in a branch, or the latest trunk delta if it has no branches. In addition, the SID you specify must not be a version currently being edited for the purpose of making a delta. To remove a delta, you must either own the SCCS file and the directory, or you must have created the delta you want to remove.

If you specify a directory for the *File* parameter, the **rmdel** command performs the requested actions on all SCCS files (those with the **s.** prefix). If you specify a **-** (dash) for the *File* parameter, the **rmdel** command reads standard input and interprets each line as the name of an SCCS file. The **rmdel** command continues to read input until it reaches an end-of-file character.

After a delta has been removed, it is not included in any g-file created by the **get** command. However, the delta table entry remains in the **s.** file with an **R** by the entry to show that the delta has been removed.

Flags

-r *SID* Removes the specified delta *SID* from the SCCS file. This flag is required.

Exit Status

This command returns the following exit values:

0 Successful completion.
>0 An error occurred.

Examples

To remove delta 1.3 from the **s.test.c** SCCS file, type:

```
rmdel -r 1.3 s.test.c
```

Files

/usr/bin/rmdel	Contains the **rmdel** command.
s.*files*	Files processed by the **rmdel** command.

Related Information

The **delta** command, **get** command, **prs** command.

The **sccsfile** file format.

Source Code Control System (SCCS) Overview in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

List of SCCS Commands in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

rmdev Command

Purpose

Removes a device from the system.

Syntax

rmdev -l *Name* [**-d** | **-S**] [**-f** *File*] [**-h**] [**-q**] [**-R**]

Description

The **rmdev** command unconfigures or both unconfigures and undefines the device specified with the device logical name (the **-l** *Name* flag). The default action unconfigures the device while retaining its device definition in the Customized Devices object class.

If you specify the **-S** flag, the **rmdev** command sets the device to the Stopped state for devices that support the Stopped state. If you specify the **-d** flag, the **rmdev** command deletes the device definition from the Customized Devices object class (undefines). If you do not specify the **-d** flag, the **rmdev** command sets the device to the Defined state (unconfigures). If you specify the **-R** flag, the **rmdev** command acts on any children of the device as well.

Attention: To protect the Configuration database, the **rmdev** command is not interruptible. Stopping this command before execution is complete could result in a corrupted database.

You can use the Devices application in Web-based System Manager to change device characteristics.

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

Flags

-d	Removes the device definition from the Customized Devices object class. This flag cannot be used with the **-S** flag.
-f *File*	Reads the needed flags from the *File* parameter.
-h	Displays the command usage message.
-l *Name*	Specifies the logical device, indicated by the *Name* parameter, in the Customized Devices object class.
-q	Suppresses the command output messages from standard output and standard error.
-R	Unconfigures the device and its children. When used with the **-d** or **-S** flags, the children are undefined or stopped, respectively.
-S	Makes the device unavailable by only calling the Stop method if the device has a Stop method. This flag cannot be used with the **-d** flag.

Security

Privilege Control: Only the root user and members of the system group should have execute (x) access to this command.

Auditing Events:

Event	Information
DEV_Stop	Device name
DEV_Unconfigure	Device name
DEV_Remove	Device name

Examples

1. To unconfigure the `cd0` CD-ROM device while retaining its device definition in the Customized Devices object class, type:

   ```
   rmdev -l cd0
   ```

   The system displays a message similar to the following:

   ```
   cd0 defined
   ```

2. To remove the `cd0` CD-ROM device definition from the Customized Devices object class, type:

   ```
   rmdev -d -l cd0
   ```

   The system displays a message similar to the following:

   ```
   cd0 deleted
   ```

3. To make Transmission Control Protocol/Internet Protocol (TCP/IP) unavailable by invoking the Stop method, type:

   ```
   rmdev -l ent0 -S
   ```

   The system displays a message similar to the following:

   ```
   ent0 stopped
   ```

Files

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

Related Information

The **chdev** command, **lsattr** command, **lsconn** command, **lsdev** command, **lsparent** command, **mkdev** command.

Device Overview for System Management in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.2 Web-based System Manager Administration Guide*.

---

rmdir Command

Purpose

Removes a directory.

Syntax

rmdir [**-p**] *Directory ...*

Description

The **rmdir** command removes the directory, specified by the *Directory* parameter, from the system. The directory must be empty before you can remove it, and you must have write permission in its parent directory. Use the **ls -al** command to check whether the directory is empty.

Note: The **rmdir** command supports the **—** (dash, dash) parameter as a delimiter that indicates the end of the flags.

Flags

-p*Directory* Removes all directories along the path name specified by the *Directory* parameter. Parent directories must be empty and the user must have write permission in the parent directories before they can be removed.

Exit Status

This command returns the following exit values:

0 Each directory entry specified by a *Directory* parameter was removed successfully.
>0 An error occurred.

Examples

1. To empty and remove a directory, type:

   ```
   rm mydir/* mydir/.*
   rmdir mydir
   ```

   This command removes the contents of the **mydir** file and then removes the empty directory. The **rm** command displays an error message about trying to remove the directories **.** (dot) and **..** (dot, dot), and then the **rmdir** command removes them.

   Note that the **rm mydir/* mydir/.*** command first removes files with names that do not begin with a dot, and then removes those with names that do begin with a dot. You may not realize that the directory contains file names that begin with a dot because the **ls** command does not usually list them unless you use the **-a** flag.

2. To remove the **/home**, **/home/demo**, and **/home/demo/mydir** directories, type:

   ```
   rmdir -p /home/demo/mydir
   ```

   This command removes first the **/mydir** directory and then the **/demo** and **/home** directories, respectively. If a directory is not empty or does not have write permission when it is to be removed, the command terminates.

Files

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

Related Information

Files Overview in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices*.

Directory Overview in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices*.

File and Directory Access Modes in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices*.

The **ls** command, **mkdir** command, **rm** command.

The **rmdir** subroutine, **unlink** subroutine.

rmf Command

Purpose

Removes folders and the messages they contain.

Syntax

rmf [**+** *Folder*] [**-interactive** | **-nointeractive**]

Description

The **rmf** command deletes the messages within the specified folder and then deletes the folder. By default, the **rmf** command confirms your request before deleting a folder. If the folder contains files that are not messages, the **rmf** command does not delete the files and returns an error.

Attention: The **rmf** command irreversibly deletes messages that do not have other links.

By default, the **rmf** command removes the current folder. When the current folder is removed, **inbox** becomes the current folder. If the **+***Folder* flag is not specified, and the **rmf** command cannot find the current folder, the command requests confirmation before removing the **+inbox** folder.

The **rmf** command does not delete any folder or any messages in a folder to which you have read-only access. The **rmf** command deletes only your private sequences and your current message information from the profile.

The **rmf** command does not delete folders recursively. You cannot remove subfolders by requesting the removal of a parent folder. If you remove a subfolder, the parent of that folder becomes the current folder.

Flags

+*Folder*	Specifies the folder to be removed.
-help	Lists the command syntax, available switches (toggles), and version information. **Note:** For Message Handler (MH), the name of this flag must be fully spelled out.
-interactive	Requests confirmation before removing the folder. If the **+***Folder* flag is not specified, this is the default.
-nointeractive	Removes the folder and its messages without requesting confirmation. This is the default.

Profile Entries

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

`Current-Folder:`	Sets the default current folder.
`Path:`	Specifies the user's MH directory.

Examples

Attention: The **rmf** command irreversibly deletes messages that do not have other links.

1. To remove the current folder called `status`, enter:

   ```
   rmf
   ```

   The system responds with a message similar to the following:

   ```
   Remove folder "status"?
   ```

If you do want the folder removed, enter `yes`. The system responds with a message similar to the following:

```
[+inbox now current]
```

2. To remove the `meetings` folder noninteractively, enter:

```
rmf +meetings
```

Files

$HOME/.mh_profile	Defines the MH user profile.
/usr/bin/rmf	Contains the **rmf** command.

Related Information

The **rmm** command.

The **mh_alias** file format, **mh_profile** file format.

Mail Overview in *AIX 5L Version 5.2 System User's Guide: Communications and Networks*.

rmfilt Command

Purpose

Removes a filter rule from the filter table.

Syntax

rmfilt -v 4|6 -n fid | all [**-f**]

Description

Use the **rmfilt** command to remove filter rules from the filter rule table. Actions by this command will not effect the IP Security subsystem until the **mkfilt** command is executed.

The **rmfilt** command removes a filter rules from the filter rule table. Only manual filter rules can be removed.

Flags

-f	Force to remove auto-generated filter rules. **-f** flag works with **-n all** to remove all the filter rules (user-defined and auto-generated filter rules) except rule number 1 for IP version 4.
-n	The ID of the filter rule you want to remove from the filter rule table. For IP version 4, the value of **1** is invalid for this flag, that is a reserved filter rule. If **all** is specified, all the user defined filter rules will be removed until the **-f** flag is specified.
-v	IP version of the filter rule you want to remove. Value **4** specifies IP version 4. Value **6** specifies IP version 6.

rmfs Command

Purpose

Removes a file system.

Syntax

rmfs [**-r** | **-i**] *FileSystem*

Description

The **rmfs** command removes a file system. If the file system is a journaled file system (JFS or JFS2), the **rmfs** command removes both the logical volume on which the file system resides and the associated stanza in the **/etc/filesystems** file. If the file system is not a JFS or JFS2 file system, the command removes only the associated stanza in the **/etc/filesystems** file. The *FileSystem* parameter specifies the file system to be removed.

You 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 rmfs** fast path to run this command.

Flags

-r Removes the mount point of the file system.
-i Displays warning and prompts the user before removing file system.

Exit Status

This command returns the following exit values:

0	File system is successfully removed.
>0	File system is not successfully removed.

Examples

To remove the **/test** file system, enter:

```
rmfs /test
```

This removes the **/test** file system, its entry in the **/etc/filesystems** file, and the underlying logical volume.

Files

/etc/rmfs	Contains the **rmfs** command.
/etc/filesystems	Lists the known file systems and defines their characteristics.

Related Information

The **chfs** command, **crfs** command, **lsfs** command, **mkfs** command, **rmlv** command.

File Systems in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices*.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.2 Web-based System Manager Administration Guide*.

System Management Interface Tool in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*.

rmgroup Command

Purpose
Removes a group.

Syntax
rmgroup [**-R** *load_module*] *Name*

Description
The **rmgroup** command removes a group specified by the *Name* parameter. This command deletes all the group attributes as well. To remove a group, the group name must already exist as a string of 8 bytes or less. Users who are group members are not removed from the system.

If the group is the primary group for any user, you cannot remove it unless you redefine the user's primary group with the **chuser** command. The **chuser** command alters the **/etc/passwd** file. Only the root user or a user with GroupAdmin authorization can remove an administrative group or a group with administrative users as members.

For groups that were created with an alternate Identification and Authentication (I&A) mechanism, the **-R** flag can be used to specify the I&A load module used. Load modules are defined in the **/usr/lib/security/methods.cfg** file.

You 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 rmgroup** fast path to run this command.

Flag
-R *load_module* Specifies the loadable I&A module used to remove a group.

Security
Access Control: This command should grant execute (x) access only to the root user and members of the security group. This command should be installed as a program in the trusted computing base (TCB). The command should be owned by the root user with the **setuid** (SUID) bit set.

Files Accessed:

Mode	File
r	**/etc/passwd**
rw	**/etc/group**
rw	**/etc/security/group**

Auditing Events:

Event	Information
GROUP_Remove	group

Limitations
Removing a group may not be supported by all loadable I&A modules. If the loadable I&A module does not support removing a group, an error is reported.

Examples

1. To remove the finance group, type:

   ```
   rmgroup finance
   ```

2. To remove the LDAP I&A loadable module group monsters, type:

   ```
   rmgroup -R LDAP monsters
   ```

Files

/usr/sbin/rmgroup	Contains the **rmgroup** command.
/etc/group	Contains the basic attributes of groups.
/etc/security/group	Contains the extended attributes of groups.

Related Information

The **chfn** command, **chgrpmem** command, **chsh** command, **chgroup** command, **chuser** command, **lsgroup** command, **lsuser** command, **mkgroup** command, **mkuser** command, **passwd** command, **pwdadm** command, **rmuser** command, **setgroups** command, **setsenv** command.

Standalone System Security in *AIX 5L Version 5.2 Security Guide*.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.2 Web-based System Manager Administration Guide*.

---

rmitab Command

Purpose

Removes records in the **/etc/inittab** file.

Syntax

rmitab *Identifier*

Description

The **rmitab** command removes an **/etc/inittab** record. You can specify a record to remove by using the *Identifier* parameter. The *Identifier* parameter specifies a field of one to fourteen characters used to uniquely identify an object. If the *Identifier* field is not unique, the command is unsuccessful.

Examples

To remove the tty entry for tty2 , enter:

```
rmitab "tty002"
```

Related Information

The **smit** command, **chitab** command, **lsitab** command, **mkitab** command, **init** command, **/etc/inittab** file.

---

rmkeyserv Command

Purpose

Stops the **keyserv** daemon and comments the entry in the **/etc/rc.nfs** file.

Syntax

/usr/sbin/rmkeyserv [**-I** | **-B** | **-N**]

Description

The **rmkeyserv** command comments the entry for the **keyserv** daemon in the **/etc/rc.nfs** file. The **rmkeyserv** daemon stops the **keyserv** daemon by using the **stopsrc** command.

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 rmkeyserv** fast path to run this command.

Flags

-I	Comments the entry for the **keyserv** daemon in the **/etc/rc.nfs** file.
-B	Comments the entry for the **keyserv** daemon in the **/etc/rc.nfs** file and stops the **keyserv** daemon. This flag is the default.
-N	Stops the **keyserv** daemon using the **stopsrc** command. This flag does not change the **/etc/rc.nfs** file.

Examples

To comment the entry in the **/etc/rc.nfs** file that starts the **keyserv** daemon, enter:

```
rmkeyserv -I
```

This command will not stop the currently executing daemon.

Files

/etc/rc.nfs	Contains the startup script for the NFS and NIS daemons.

Related Information

The **smit** command.

The **keyserv** daemon.

Network File System (NFS) Overview for System Management in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

How to Start the NFS Daemons, How to Stop the NFS Daemons, in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

Exporting a File System Using Secure NFS and Mounting a File System Using Secure NFS in *AIX 5L Version 5.2 Security Guide*.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.2 Web-based System Manager Administration Guide*.

Network Information Service (NIS) in *AIX 5L Version 5.2 Network Information Services (NIS and NIS+) Guide*.

NIS Reference.

rmlv Command

Purpose

Removes logical volumes from a volume group.

Syntax

rmlv [**-B**] [**-f**] [**-p** *Physical Volume*] *LogicalVolume ...*

Description

Attention: This command destroys all data in the specified logical volumes.

The **rmlv** command removes a logical volume. The LogicalVolume parameter can be a logical volume name or logical volume ID. The logical volume first must be closed. If the *volume group* is varied on in concurrent mode, the logical volume must be closed on all the concurrent nodes on which *volume group* is varied on. For example, if the logical volume contains a file system, it must be unmounted. However, removing the logical volume does not notify the operating system that the file system residing on it have been destroyed. The command **rmfs** updates the **/etc/filesystems** file.

Notes:

1. To use this command, you must either have root user authority or be a member of the **system** group.
2. The **rmlv** command is not allowed on a snapshot volume group or a volume group that has a snapshot volume group.

You can use the Volumes application in Web-based System Manager (wsm) to change volume characteristics.

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

Flags

-B	Issues a **chlvcopy -B -s** for the parent logical volume if the logical volume was created using the **-l** flag. If it is a regular logical volume then the **-B** flag is ignored.
-f	Removes the logical volumes without requesting confirmation.
-p *PhysicalVolume*	Removes only the logical partition on the *PhysicalVolume*. The logical volume is not removed unless there are no other physical partitions allocated.

Examples

Attention: The command used in this example destroys all data in the logical volumes.

To remove logical volume `lv05` without requiring user confirmation, enter the following command:

```
rmlv  -f lv05
```

The logical volume is removed from the volume group.

Files

/etc	Directory where the **rmlv** command resides.
/tmp	Directory where the temporary files are stored while the command is running.
/etc/filesystems	Lists the known file systems and defines their characteristics.

Related Information

The **varyonvg** command, **unmount** command, **rmfs** command.

The Logical Volume Storage Overview in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices* explains the Logical Volume Manager, physical volumes, logical volumes, volume groups, organization, ensuring data integrity, and allocation characteristics.

The System Management Interface Tool in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices* 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 5.2 Web-based System Manager Administration Guide*.

---

rmlvcopy Command

Purpose

Removes copies from a logical volume.

Syntax

rmlvcopy *LogicalVolume Copies* [*PhysicalVolume* ...]

Description

The **rmlvcopy** command removes copies from each logical partition in the *LogicalVolume*. Copies are the physical partitions which, in addition to the original physical partition, make up a logical partition. You can have up to two copies in a logical volume. The *Copies* parameter determines the maximum number of physical partitions that remain. The *LogicalVolume* parameter can be a logical volume name or logical volume ID. The *PhysicalVolume* parameter can be the physical volume name or the physical volume ID. If the *PhysicalVolume* parameter is used, then only copies from that physical volume will be removed.

You can use the Volumes application in Web-based System Manager (wsm) to change volume characteristics.

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

Notes:

1. To use this command, you must either have `root` user authority or be a member of the **system** group.
2. If LVM has not recognized that a disk has failed it is possible that LVM will remove a different mirror. Therefore if you know that a disk has failed and LVM does not show those disks as missing you should specify the failed disks on the command line or you should use **replacepv** to replace the disk or **reducevg** to remove the disk.
3. The **rmlvcopy** command is not allowed on a snapshot volume group or a volume group that has a snapshot volume group.

Examples

To reduce the number of copies of each logical partition belonging to logical volume `lv0112`, enter:

```
rmlvcopy lv0112 2
```

Each logical partition in the logical volume now has at most two physical partitions.

Files

/etc/rmlvcopy	Contains the **rmlvcopy** command.
/tmp/*	Directory where the temporary files are stored while the command is running.

Related Information

The **mklv** command, **mklvcopy** command.

The Logical Volume Storage Overview in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices* 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 5.2 Web-based System Manager Administration Guide*.

The System Management Interface Tool in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices* explains the structure, main menus, and tasks that are done with SMIT.

---

rmm Command

Purpose

Removes messages from active status.

Syntax

rmm [**+** *Folder*] [*Messages*]

Description

The **rmm** command removes messages from active status by renaming them. To rename a message, the system prefaces the current message number with a , (comma). Inactive files are unavailable to the Message Handler (MH) package. However, system commands can still manipulate inactive files.

Note: The **rmm** command does not change the current message.

Inactive messages should be deleted periodically. An entry can be placed in your **crontab** file to automatically delete all files beginning with a comma.

Flags

+*Folder*	Specifies the folder containing the messages to rename.

Messages Specifies the messages to rename. You can specify several messages, a range of messages, or a single message. Use the following references to specify a message:

Number
Number of the message

Sequence
A group of messages specified by the user. Recognized values include:

all All messages in a folder

cur or . (dot)
Current message. This is the default.

first First message in a folder

last Last message in a folder

next Message following the current message

prev Message preceding the current message

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

Examples

1. To remove the current message in the current folder, enter:
   ```
   rmm
   ```
2. To remove messages `2` through `5` from the `sales` folder, enter:

   ```
   rmm  +sales 2-5
   ```

Profile Entries

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

`Current-Folder:`	Sets the default current folder.
`Path:`	Specifies the *UserMhDirectory*.
`rmmproc:`	Specifies the program used to remove messages from a folder.

Files

$HOME/.mh_profile	Contains the MH user profile.
/usr/bin/rmm	Contains the **rmm** command.

Related Information

The **crontab** command, **rmf** command.

The **.mh_alias** file format, **.mh_profile** file format.

Mail Overview in *AIX 5L Version 5.2 System User's Guide: Communications and Networks*.

---

rmnamsv Command

Purpose

Unconfigures TCP/IP-based name service on a host.

Syntax

rmnamsv [**-f** | **-F** *FileName*]

Description

The **rmnamsv** high-level command unconfigures a TCP/IP-based name service on a host. You can unconfigure name service for a host functioning as a client.

To unconfigure name service for a client, the **rmnamsv** command calls the **namerslv** low-level command to unconfigure entries in the **/etc/resolv.conf** file or to rename the **/etc/resolv.conf** file to a default or user-specified file name.

You can use the Network application in Web-based System Manager to change network characteristics.

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

Flags

-F *FileName*	Renames the system configuration database to the file name specified by *FileName*.
-f	Specifies that the default file name (**/etc/resolv.conf.sv**) should be used to rename the **/etc/resolv.conf** file.

Files

/etc/resolv.conf	Contains the default system configuration database.

Related Information

Naming in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

Understanding the SMIT Interface for TCP/IP in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.2 Web-based System Manager Administration Guide*.

The **namerslv** command.

---

rmnfs Command

Purpose

Changes the configuration of the system to stop running NFS daemons.

Syntax

/usr/sbin/rmnfs [**-I** | **-N** | **-B**]

Description

The **rmnfs** command changes the current configuration of the system so that the **/etc/rc.nfs** file is not executed on system restart. In addition, you can direct the command to stop NFS daemons that are currently running.

Flags

-B Removes the entry in the **inittab** file and stops NFS daemons that are currently executing. This flag is the default.
-I Removes the entry in the **inittab** file that starts NFS daemons on system restart.
-N Stops immediately NFS daemons and does not change the **inittab** file.

Examples

To stop all of the NFS daemons immediately, enter:

```
rmnfs -N
```

This command will not change the **inittab** file.

Related Information

The **chnfs** command, **mknfs** command, **rmnfsexp** command, **rmnfsmnt** command.

List of NFS Commands.

Network File System (NFS) Overview for System Management in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

NFS Installation and Configuration in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

---

rmnfsexp Command

Purpose

Unexports a directory from NFS clients.

Syntax

/usr/sbin/rmnfsexp -d *Directory* [**-f** *Exports_file*] [**-I** | **-B** | **-N**]

Description

The **rmnfsexp** command removes an entry from the exports list for NFS clients. The **rmnfsexp** command starts the **exportfs** command to unexport the specified directory. If an entry exists in the **/etc/exports** file, that entry is removed.

Flags

-d *Directory*	Specifies the directory to be unexported.
-f *Exports_File*	Specifies the full pathname of the exports file to use if other than the **/etc/exports** file.
-I	This flag directs the command to remove the entry from the **/etc/exports** file without executing the **exportfs** command.
-B	Removes the entry in the **/etc/exports** file for the directory specified and executes the **exportfs** command to remove the export.
-N	Unexports the directory immediately by invoking the **exportfs** command. The **/etc/exports** file is not modified with this flag.

Examples

1. To unexport a directory immediately, type:

   ```
   rmnfsexp -d /usr -N
   ```

   In this example, the **/usr** directory is unexported immediately.
2. To unexport a directory immediately and after every system restart, type:

   ```
   rmnfsexp -d /home/guest -B
   ```
3. To unexport a directory immediately from an exports file other than the **/etc/exports** file, type:

   ```
   rmnfsexp -d /usr -f /etc/exports.other -N
   ```

Files

/etc/exports	Lists the directories that the server can export.
/etc/xtab	Lists currently exported directories.
/etc/netgroup	Contains information about each user group on the network.

Related Information

The **chnfsexp** command, **exportfs** command, **mknfsexp** command.

List of NFS Commands.

Network File System (NFS) Overview for System Management in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

---

rmnfsmnt Command

Purpose

Removes an NFS mount.

Syntax

/usr/sbin/rmnfsmnt -f *PathName* [**-l** | **-B** | **-N**]

Description

The **rmnfsmnt** command removes the appropriate entry from the **/etc/filesystems** file and unmounts the file system specified. When used with the **-N** flag, the **rmnfsmnt** command unmounts the file system and does not modify the **/etc/filesystems** file.

Flags

-B	Removes the entry in the **/etc/filesystems** file and unmounts the directory. If no entry exists in the **/etc/filesystems** file, the flag makes no changes to the file. If the file system is not currently mounted, the flag does not attempt to unmount it. This flag is the default.
-f *PathName*	Specifies the path name of the NFS-mounted file system.
-l	Removes the entry specified by the path name from the **/etc/filesystems** file.
-N	Unmounts the specified directory and does not modify the **/etc/filesystems** file.

Examples

1. To unmount a file system, enter:

   ```
   rmnfsmnt -f /usr/man -N
   ```

In this example, the /usr/man file system is unmounted.

2. To remove a mount for a file, enter:

```
rmnfsmnt -f /usr/local/man -B
```

In this example, the mount for the /usr/local/man file is removed.

File

/etc/filesystems	Lists the remote file systems to mount during the system restart.

Related Information

The **chnfsmnt** command, **mknfsmnt** command, **mount** command, **umount** command.

Network File System (NFS) Overview for System Management in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

List of NFS Commands.

---

rmnotify Command

Purpose

Removes a notify method definition from the Notify object class.

Syntax

rmnotify -n *NotifyName*

Description

The **rmnotify** command removes a notify method definition from the notify object class.

Flags

-n *NotifyName*	Specifies the notify method definition to be removed. The **rmnotify** command is unsuccessful if the *NotifyName* name does not already exist in the Notify object class.

Files

/etc/objrepos/SRCnotify	Specifies the SRC Notify Method object class.

Related Information

The **lssrc** command, **mknotify** command, **mkssys** command, **rmssys** command.

System Resource Controller Overview in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*.

System Resource Controller (SRC) Overview for Programmers in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

Understanding SRC Objects in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

rmpath Command

Purpose

Removes from the system a path to an MPIO capable device.

Syntax

rmpath [**-l** *Name*] [**-p** *Parent*] [**-w** *Connection*] [**-d**]

rmpath -h

Description

The **rmpath** command unconfigures, and possibly undefines, one or more paths associated with the specified target device (**-l** *Name*). The set of paths that are removed aredetermined by the combination of the **-l** *Name*, **-p** *Parent* and **-w** *Connection* flags. If the command will result in all paths associated with the device being unconfigured or undefined, the command will exit with an error and without unconfiguring or undefining any path. In this situation, **rmdev** command must be used instead to unconfigure or undefine the target device itself.

The default action unconfigures each specified path, but does not completely remove it from the system. If the **-d** flag is specified, the **rmpath** command unconfigures (if necessary) and removes, or deletes, the path definition(s) from the system.

When the **rmpath** command finishes, it displays a status message. When unconfiguring paths, it is possible for this command to be able to unconfigure some paths and not others (e.g., paths that are in the process of doing I/O cannot be unconfigured).

The **rmpath** command provides status messages about the results of operation. Messages in one of the following formats will be generated:

path [defined | deleted]
: This message is displayed when a single path was successfully unconfigures or undefined. If the path is successfully configured the message `path available` displays. If the path is not successfully configured and there is no explicit error code returned by the method, the message `path defined` displays.

paths [defined | deleted]
: This message is displayed if multiple paths were identified and all paths were successfully unconfigured or undefined. If the **-d** flag is not specified, the message would be `paths defined`. If the **-d** flag is specified, the message would be `paths deleted`.

some paths [defined | deleted]
: This message is display if multiple paths were identified, but only some of them were successfully unconfigured or undefined. If the **-d** flag is not specified, the message would be `some paths defined`. If the '**-d** flag is specified, the message would be `some paths deleted`.

no paths processed
: This message is generated if no paths were found matching the selection criteria.

Flags

-d	Indicates that the specified paths are to be deleted from the system.
-h	Displays the command usage message.
-l *Name*	Specifies the logical device name of the target device whose path is to be removed. The paths to be removed are qualified via the **-p** and **-w** flags.

-p *Parent* Indicates the logical device name of the parent device to use in qualifying the paths to be removed. Since all paths to a device cannot be removed by this command, either this flag, the **-w** flag, or both must be specified.

-w *Connection* Indicates the connection information to use in qualifying the paths to be removed. Since all paths to a device cannot be removed by this command, either this flag, the **-p** flag, or both must be specified.

Security

Privilege Control: Only the root user and members of the system group have execute access to this command.

Auditing Events:

Event	Information
DEV_Change	rmpath,Unconfigure,<unconfigure method arguments>
DEV_Change	rmpath,Undefine,<undefine method arguments>

Examples

1. To unconfigure the path from **scsi0** to **hdisk1** at connection **5,0**, type:

   ```
   rmpath -l hdisk1 -p scsi0 -w "5,0"
   ```

   The message generated would be similar to:

   ```
   path defined
   ```

2. To unconfigure all paths from **scsi0** to **hdisk1**, type:

   ```
   rmpath -l hdisk1 -p scsi0
   ```

   If all paths were successfully unconfigured, the message generated would be similar to:

   ```
   paths defined
   ```

   However, if only some of the paths were successfully unconfigured, the message would be similar to:

   ```
   some paths defined
   ```

3. To undefine the path definition between **scsi0** and **hdisk1** at connection **5,0**, type:

   ```
   rmpath -d -l hdisk1 -p scsi0 -w "5,0"
   ```

   The message generated would be similar to the following:

   ```
   path deleted
   ```

4. To unconfigure all paths from **scsi0** to **hdisk1**, type:

   ```
   rmpath -d -l hdisk1 -p scsi0
   ```

   The message generated would be similar to:

   ```
   paths deleted
   ```

Files

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

Related Information

The **chpath** command, **lspath** command, **mkpath** command.

---

rmprtsv Command

Purpose

Unconfigures a print service on a client or server machine.

Syntax

rmprtsv { **-c** | **-s** } [**-T** | **-U** | **-A**] [**-h** *"HostName ..."* | **-H** *FileName*] [**-q** *"QEntry ..."*] [**-q** *QEntry* **-v** *"DeviceName ..."*]

Description

The **rmprtsv** high-level command unconfigures a print service on a client or server machine.

To unconfigure print service for a client, the **rmprtsv** command calls the **rmque** and **rmquedev** commands to disable the client spool queue and to remove the appropriate entries in the **/etc/qconfig** file.

To unconfigure print service for a server, the **rmprtsv** command performs the following procedure:

1. Calls the **stopsrc** command to deactivate the **lpd** and **qdaemon** servers.
2. Calls the **ruser** low-level command to unconfigure remote users on the print server.
3. Calls the **rmque** and **rmquedev** commands to unconfigure the spooler and its device queues, and delete the appropriate entries in the server's **/usr/lib/lpd/qconfig** file.

Flags

-A	Removes specified entries from the **/etc/qconfig** file but does not fully unconfigure print service.
-c	Unconfigures print service for a client machine. Use the **-q** flag with the **-c** flag.
-H *FileName*	Specifies the name of a file containing a list of host names to be left configured for print service.
-h *"HostName..."*	Specifies a list of remote host names not allowed to use the print server. Note that the queuing system does not support multibyte host names.
-q *"QEntry..."*	Specifies a list of entries to remove from the **/etc/qconfig** file.
-s	Unconfigures print service for a server machine. The **-h**, **-H**, and **-q** flags should be used with the **-s** flag.
-T	Stops print service but does not fully unconfigure print service.
-U	Removes specified remote users on the print server but does not fully unconfigure print service.
-v *"DeviceName..."*	Specifies a list of the names of the device stanzas in the **qconfig** file. Must be used with the **-q** *QEntry* flag.

Files

/etc/qconfig	Contains configuration information for the printer queueing system.

Related Information

The **rmque** command, **rmquedev** command, **ruser** command.

The **lpd** daemon, **qdaemon** daemon.

The SMIT Interface for TCP/IP in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

rmps Command

Purpose

Removes an inactive paging space.

Syntax

rmps *PagingSpace*

Description

The **rmps** command removes an inactive paging space. The *PagingSpace* parameter specifies the name of the paging space to be removed, which is actually the name of the logical volume on which the paging space resides.

For an NFS paging space, the *PagingSpace* parameter specifies the name of the paging space to be removed. The device and its definition, which corresponds to this paging space, will be removed from the system. Nothing will be changed on the NFS server where the file that is used for paging actually resides.

Active paging spaces cannot be removed. In order to remove an active paging space, change its characteristics so that it is not configured at subsequent system restarts (the **chps** command). Then, on the next system restart, the paging space is not active, and it can be removed.

You can use the File Systems application in Web-based System Manager (wsm) to change file system characteristics.

Examples

To remove `PS01` paging space, enter:

```
rmps PS01
```

This removes the `PS01` paging space.

Files

/etc/swapspaces	Specifies the paging space devices activated by the **swapon -a** command.

Related Information

The **lsps** command, **mkps** command, **chps** command, **rmlv** command, **swapon** command.

The Paging Space Overview in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices* explains paging space and its allocation policies.

The File Systems in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices* explains file system types, management, structure, and maintenance.

The Logical Volume Storage Overview in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices* 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 5.2 Web-based System Manager Administration Guide*.

The System Management Interface Tool in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices* explains the structure, main menus, and tasks that are done with SMIT.

rmqos Command

Purpose

Changes the configuration of the system to remove QoS support.

Syntax

/usr/sbin/rmqos [**-I** | **-N** | **-B**]

Description

The **rmqos** command changes the current configuration of the system to remove Quality of Service (QoS) support.

Flags

-B	Removes the entry in the **inittab** file that enables QoS at system startup and stops the QoS daemons. This flag is the default.
-I	Removes the entry in the **inttab** file that enables QoS at system startup but does not affect the currently running QoS subsystem.
-N	Disables QoS support immediately but does not change the **inittab** file.

Files

inittab	Controls the initialization process of the system.
/etc/rc.qos	Contains the startup script for the QoS daemons.

Related Information

The **mkqos** command.

TCP/IP Quality of Service (QoS) in the *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

rmque Command

Purpose

Removes a printer queue from the system.

Syntax

rmque -q *Name*

Description

The **rmque** command removes a queue from the system configuration by deleting the queue stanza named by the **-q** flag from the **/etc/qconfig** file. All queue devices must be deleted using the **rmquedev** command before entering this command.

You 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 rmque** fast path to run this command.

> **Note:** Do not edit the **/etc/qconfig** file while there are active jobs in any queue. Editing includes both manual editing and use of the **chque**, **mkque**, **rmque**, **mkquedev**, **rmquedev**, or **chquedev** commands. It is recommended that all changes to the **/etc/qconfig** file be made using these commands. However, if manual editing is desired, first issue the **enq -G** command to bring the queuing system and the qdaemon to a halt after all jobs are processed. Then edit the **/etc/qconfig** file and restart the qdaemon with the new configuration.

Flags

-q *Name* Specifies the name of the queue to be removed.

Examples

To remove printer queue `lp0`, enter:

```
rmque -q lp0
```

Files

/usr/bin/rmque	Contains the **rmque** command.
/etc/qconfig	Contains the configuration file.

Related Information

The **chque** command, **lsque** command, **mkque** command, **rmquedev** command, **smit** command.

The **/etc/qconfig** file.

Printer Overview for System Management in *AIX 5L Version 5.2 Guide to Printers and Printing*.

Spooler Overview for System Management in *AIX 5L Version 5.2 Guide to Printers and Printing*.

Removing a Print Queue in *AIX 5L Version 5.2 Guide to Printers and Printing*.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.2 Web-based System Manager Administration Guide*.

---

rmquedev Command

Purpose

Removes a printer or plotter queue device from the system.

Syntax

rmquedev -d *Name* **-q** *Name*

Description

The **rmquedev** command removes a printer or plotter queue device from the system configuration by deleting the device stanza named by the **-d** flag from the **/etc/qconfig** file. It also modifies the `Device=DeviceName1,DeviceName2,DeviceName3` line of the queue stanza, deleting the entry for the device `Name`.

You 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 rmquedev** fast path to run this command.

> **Note:** Do not edit the **/etc/qconfig** file while there are active jobs in any queue. Editing includes both manual editing and use of the **chque**, **mkque**, **rmque**, **mkquedev**, **rmquedev**, or **chquedev** commands. It is recommended that all changes to the **/etc/qconfig** file be made using these commands. However, if manual editing is desired, first issue the **enq -G** command to bring the queuing system and the qdaemon to a halt after all jobs are processed. Then edit the **/etc/qconfig** file and restart the qdaemon with the new configuration.

Flags

-d *Name* Specifies the *Name* of the device stanza to be deleted from the **qconfig** file.
-q *Name* Specifies the *Name* of the device to be modified in the preceding queue stanza.

Examples

To delete the `loc` device stanza from the **/etc/qconfig** file and modify the `"DEVICE ="` stanza in the preceding queue stanza `lpq`, enter:

```
rmquedev  -q lpq  -d loc
```

Files

/usr/bin/rmquedev	Contains the **rmquedev** command.
/etc/qconfig	Configuration file.

Related Information

The **chquedev** command, **lsquedev** command, **mkquedev** command, **rmque** command, **smit** command.

The **/etc/qconfig** file.

Printer Overview for System Management in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Spooler Overview for System Management in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Removing a Print Queue in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.2 Web-based System Manager Administration Guide*.

---

rmramdisk Command

Purpose

Removes RAM disks created by the **mkramdisk** command.

Syntax

rmramdisk *ram_disk_name*

Description

The **rmramdisk** command removes the specified RAM disk and the device special files created for that RAM disk. RAM disks are also removed when the system is rebooted.

Parameters

ram_disk_name Name of the specific RAM disk to be removed from memory. If not specified, an error is returned. The names of the RAM disks are in the form of **rramdisk***x* where *x* is the logical RAM disk number (0 through 63).

Exit Status

This command returns the following exit values:

0 Successful completion.
>0 An error occurred.

Examples

```
# ls -l /dev/*ramdisk2
brw-------   1 root     system       38,  0 Aug 01 05:52 /dev/ramdisk2
crw-------   1 root     system       38,  0 Aug 01 05:52 /dev/rramdisk2
```

To remove the ramdisk2, enter:

```
# rmramdisk ramdisk2

# ls -l /dev/*ramdisk2
ls: 0653-341 The file /dev/*ramdisk2 does not exist.
```

Files

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

Related Information

The **mkramdisk** command, and **ls** command.

---

rmresponse Command

Purpose

Removes a response.

Syntax

rmresponse [**–f**] [**–q**] [**–h**] [**–TV**] *response*[**:***node_name*]

Description

The **rmresponse** command removes the response specified by the *response* parameter. The response must already exist in order to be removed. When the response must be removed even if it is linked with conditions, specify the **-f** flag. This forces the response and the links with the conditions to be removed. If the **-f** flag is not specified and links with conditions exist, the response is not removed. This command does not remove conditions.

Flags

–f Forces the response to be removed even if it is linked with conditions. The links with the conditions are removed as well as the response, but the conditions are not removed.

–q Does not return an error when *Response* does not exist.

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

–T Writes the command's trace messages to standard error. For your software service organization's use only.

–V Writes the command's verbose messages to standard output.

Parameters

response Specifies the name of a defined response to be removed.

node_name Specifies the node in a domain where the response is defined. If *node_name* is not specified, the local node is used. *node_name* is a node within the scope determined by the CT_MANAGEMENT_SCOPE environment variable.

Security

The user needs write permission for the **IBM.EventResponse** resource class to run **rmresponse**. Permissions are specified in the access control list (ACL) file on the contacted system. See the *RSCT Guide and Reference* 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_MANAGEMENT_SCOPE

Determines the management scope that is used for the session with the RMC daemon in processing the resources of the event response resource manager (ERRM). The management scope determines the set of possible target nodes where the resources can be processed. The valid values are:

0 Specifies *local* scope.

1 Specifies *local* scope.

2 Specifies *peer domain* scope.

3 Specifies *management domain* scope.

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

Standard Output

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

Standard Error

All trace messages are written to standard error.

Examples

These examples apply to standalone systems:

1. To remove the response definition named ″Broadcast event on-shift″, run this command:

```
rmresponse  "Broadcast event on-shift"
```

2. To remove the response definition named ″Broadcast event on-shift″ even if the response is linked with conditions, run this command:

```
rmresponse -f "Broadcast event on-shift"
```

These examples apply to management domains:

1. In this example, the current node is the management server. To remove the response definition named ″Broadcast event on-shift″ on managed node **nodeB**, run this command:

```
rmresponse  "Broadcast event on-shift":nodeB
```

2. To remove the response definition named ″Broadcast event on-shift″ defined on node **nodeA**, run this command from any node in the domain:

```
rmresponse  "Broadcast event on-shift":nodeA
```

Location

/usr/sbin/rsct/bin/rmresponse
Contains the **rmresponse** command

Related Information

ERRM commands: **chresponse**, **lscondresp**, **lsresponse**, **mkcondresp**, **mkresponse**

Files: **rmccli**

Books: see the *RSCT Guide and Reference* for more information about ERRM operations

---

rmrole Command

Purpose

Removes a role. This command applies only to AIX 4.2.1 and later.

Syntax

rmrole *Name*

Description

The **rmrole** command removes the role identified by the *Name* parameter from the **/etc/security/roles** file. The role name must already exist.

You can use Web-based System Manager Users application or the System Management Interface Tool (SMIT) to run this command.

Security

Files Accessed:

Mode	File
rw	/etc/security/roles
r	/etc/security/user.roles

Auditing Events:

Event	Information
ROLE_Remove	role

Examples

1. To remove the `ManageObjects` role, enter:

   ```
   rmrole ManageObjects
   ```

Files

/etc/security/roles	Contains the attributes of roles.
/etc/security/user.roles	Contains the role attribute of users.

Related Information

The **chrole** command, **lsrole** command, **mkrole** command, **chuser** command, **lsuser** command, **mkuser** command.

Standalone System Security in *AIX 5L Version 5.2 Security Guide*.

Users, Groups, and Passwords in*AIX 5L Version 5.2 Security Guide*.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.2 Web-based System Manager Administration Guide*.

---

rmrpdomain Command

Purpose

Removes a peer domain that has already been defined.

Syntax

rmrpdomain [**–f**] [**–q**] [**–h**] [**–TV**] *domain_name*

Description

The **rmrpdomain** command removes the peer domain definition specified by the *domain_name* parameter. The peer domain must already be defined to be removed. The command must be run on a node that is defined in the peer domain. When run on a node that is online to the peer domain and peer domain quorum exists, the **rmrpdomain** command removes the peer domain definition on all nodes defined to the peer domain that are reachable from that node. If a node that is defined to the peer domain is not reachable, that nodes local peer domain definition is not removed. To remove the local peer domain

definition when the peer domain is not online or when the node is not online to the peer domain, run the **rmrpdomain** command on that node and specify the **-f** flag.

The most efficient way to remove a peer domain definition is to make sure the peer domain is online. Then, from a node that is online to the peer domain, run the **rmrpdomain** command. If there are nodes that are not reachable from the node on which the **rmrpdomain** command was run, on each of those nodes, run the **rmrpdomain** command using the **-f** flag. This can be done at a later time if the node itself is not operational.

The **-f** flag must also be used to override a subsystem's rejection of the peer domain removal. A subsystem may reject the request if a peer domain resource is busy, for example. Specifying the **-f** flag in this situation indicates to the subsystems that the peer domain definition must be removed. The **-f** flag does not override the lack of peer domain quorum. An online peer domain cannot be removed if peer domain quorum does not exist.

More than half of the nodes must be online to remove a peer domain.

Flags

-f Forces the peer domain to be removed. The force flag is required to remove a peer domain definition:

- from the local node when the node is not online to the peer domain.
- when a subsystem may reject the request, as when resources are allocated, for example.

-q Specifies quiet mode. The command does not return an error if the peer domain does not exist.

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

-T Writes the command's trace messages to standard error. For your software service organization's use only.

-V Writes the command's verbose messages to standard output.

Parameters

domain_name Specifies the name of the defined peer domain that is to be removed.

Security

The user of the **rmrpdomain** command needs write permission to the **IBM.PeerDomain** resource class on each node that is to be defined to the peer domain. By default, **root** on any node in the peer domain has read and write access to this resource class through the configuration resource manager.

Exit Status

0 The command ran successfully.

1 An error occurred with RMC.

2 An error occurred with a command-line interface script.

3 An incorrect flag was entered on the command line.

4 An incorrect parameter was entered on the command line.

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

6 The peer domain definition does not exist.

Environment Variables

CT_CONTACT

Determines the system where the session with the resource monitoring and control (RMC) daemon occurs. When CT_CONTACT is set to a host name or IP address, the command contacts the RMC daemon on the specified host. If CT_CONTACT is not set, the command contacts the RMC daemon on the local system where the command is being run. The target of the RMC daemon session and the management scope determine the resource classes or resources that are processed.

Restrictions

The node on which this command is run must be defined to the peer domain and should be able to reach all of the nodes that are defined to the peer domain. The node's local peer domain definition will not be removed if the node is not reachable.

Standard Input

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

Standard Output

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

Standard Error

All trace messages are written to standard error.

Examples

1. To remove the peer domain definition of **ApplDomain** where **nodeA**, **nodeB**, and **nodeC** are defined and online to *ApplDomain*, and all are reachable to each other, run this command on *nodeA*, **nodeB**,or **nodeC**:

   ```
   rmrpdomain ApplDomain
   ```

2. To remove the local peer domain definition of **ApplDomain** on **nodeD** when **nodeD** is not online to the peer domain, the peer domain is offline, or the peer domain does not exist, run this command on **nodeD**:

   ```
   rmrpdomain -f ApplDomain
   ```

3. To remove the peer domain definition of **ApplDomain** where **nodeA**, **nodeB**, and **nodeC** are defined and online to **ApplDomain**, all are reachable to each other, and to prevent a subsystem from rejecting the request, run this command on **nodeA**, **nodeB**, or **nodeC**:

   ```
   rmrpdomain -f ApplDomain
   ```

Location

/usr/sbin/rsct/bin/rmrpdomain

Contains the **rmrpdomain** command

Files

The **/etc/services** file is modified.

Related Information

Commands: **addrpnode**, **lsrpdomain**, **lsrpnode**, **mkrpdomain**, **preprpnode**, **rmrpnode**, **startrpdomain**, **stoprpdomain**

rmrpnode Command

Purpose

Removes one or more nodes from a peer domain definition.

Syntax

rmrpnode [**–q**] [**–h**] [**–TV**] *node_name1* [*node_name2* ...]

Description

The **rmrpnode** command removes the specified nodes from the online peer domain where the command is run. The command must be run on a node that is online to the peer domain in which the nodes are to be removed. The nodes to be removed must be offline to the peer domain and must be reachable from the node that runs the command. Use the **stoprpnode** command to bring nodes offline.

More than half of the nodes must be online to remove one or more nodes from the domain.

Flags

-q Specifies quiet mode. The command does not return an error if the node is not in the peer domain.

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

-T Writes the command's trace messages to standard error. For your software service organization's use only.

-V Writes the command's verbose messages to standard output.

Parameters

node_name1 **[***node_name2* **...]**

Specifies the peer domain node name of the node to be removed from the peer domain definition. To list the peer domain node names, run the **lsrpnode** command.

Security

The user of the **rmrpnode** command needs write permission for the **IBM.PeerNode** resource class on each node that is to be removed from the peer domain. By default, **root** on any node in the peer domain has read and write access to this resource class through the configuration resource manager.

Exit Status

0 The command ran successfully.

1 An error occurred with RMC.

2 An error occurred with a command-line interface script.

3 An incorrect flag was entered on the command line.

4 An incorrect parameter was entered on the command line.

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

6 The node does not exist in the peer domain.

Environment Variables

CT_CONTACT
Determines the system where the session with the resource monitoring and control (RMC) daemon occurs. When CT_CONTACT is set to a host name or IP address, the command contacts the RMC daemon on the specified host. If CT_CONTACT is not set, the command contacts the RMC daemon on the local system where the command is being run. The target of the RMC daemon session and the management scope determine the resource classes or resources that are processed.

Restrictions

This command must be run on a node that is online in the peer domain in which the nodes are to be removed. The nodes to be removed must also be offline to the peer domain.

Standard Input

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

Standard Output

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

Standard Error

All trace messages are written to standard error.

Examples

To remove the peer domain definitions of nodes **nodeB** and **nodeC** from the peer domain **ApplDomain**, when **nodeA** is defined and online to **ApplDomain**, and **nodeB** and **nodeC** are reachable from **nodeA**, run this command from **nodeA**:

```
rmrpnode nodeB nodeC
```

Location

/usr/sbin/rsct/bin/rmrpnode Contains the **rmrpnode** command

Files

The **/etc/services** file is modified.

Related Information

Commands: **addrpnode**, **lsrpnode**, **preprpnode**, **startrpnode**, **stoprpnode**

---

rmrset Command

Purpose

Remove an rset from the system registry.

Syntax

```
rmrset rsetname
```

Description

The **rmrset** command removes an rset from the system registry.

Parameters

rsetname The name of the rset to be removed from the system registry. 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.

Security

The user must have `root` authority, or CAP_NUMA_ATTACH capability and write access permission to the specified rset.

Examples

1. To remove **test/cpus0to7** from system registry, type:

   ```
   rmrset test/cpus0to7
   ```

Files

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

Related Information

The **attachrset**, **detachrset**, **execrset**, **lsrset**, and **mkrset** commands.

---

rmrsrc Command

Purpose

Removes a defined resource.

Syntax

rmrsrc –s ″*selection_string* [**–a**] [**–h**] [**–TV**] *resource_class*

rmrsrc –r *resource_handle* [**–a**] [**–h**] [**–TV**]

Description

The **rmrsrc** command removes, or ″undefines″ the specified resource instance or resource instances. The **rmrsrc** command makes a request to resource monitoring and control (RMC) to undefine a specific resource instance. The resource manager of the resource removes the resource.

The first format of this command requires a resource class name parameter and a selection string specified using the **-s** flag. All resources in the specified resource class that match the specified selection string are removed. If the selection string identifies more than one resource to be removed, it is the same as running this command once for each resource that matches the selection string.

The second format of this command allows the actual resource handle linked with a specific resource to be specified as the parameter. It is expected that this form of the command would be more likely used from within a script.

Flags

–a Specifies that this command applies to all nodes in the cluster. The cluster scope is determined by the CT_MANAGEMENT_SCOPE environment variable. If it is not set, first the management

domain scope is chosen if it exists, then the peer domain scope is chosen if it exists, and then local scope is chosen, until the scope is valid for the command. The command will run once for the first valid scope found. For example, if both a management and peer domain exist, **rmrsrc -a** with CT_MANAGEMENT_SCOPE not set will apply to the management domain. In this case, to apply to the peer domain, set CT_MANAGEMENT_SCOPE to 2.

–r *resource_handle*
Specifies a resource handle. The resource handle must be specified using the format: *"***0x***nnnn* **0x***nnnn* **0x***nnnnnnnn* **0x***nnnnnnnn* **0x***nnnnnnnn* **0x***nnnnnnnn"*, where *n* is any valid hexadecimal digit. The resource handle uniquely identifies a particular resource instance that should be removed.

–s *"selection_string"*
Specifies a selection string. All selection strings must be enclosed within either double or single quotation marks. If the selection string contains double quotation marks, enclose the entire selection string in single quotation marks. For example:

```
-s 'Name == "testing"'
-s 'Name ?= "test"'
```

Only persistent attributes may be listed in a selection string. For information on how to specify selection strings, see the *RSCT Guide and Reference*.

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

–T Writes the command's trace messages to standard error. For your software service organization's use only.

–V Writes the command's verbose messages to standard output.

Parameters

resource_class
Specifies the resource class name. The resource instances for this resource class that match the selection string criteria are removed.

Security

The user needs write permission for the *resource_class* specified in **rmrsrc** to run **rmrsrc**. Permissions are specified in the access control list (ACL) file on the contacted system. See the *RSCT Guide and Reference* for information about the ACL file and how to modify it.

Exit Status

0 Command has run successfully.

1 Error occurred with RMC.

2 Error occurred with CLI script.

3 Incorrect flag on command line.

4 Incorrect parameter on command line.

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

0 Specifies *local* scope.

1 Specifies *local* scope.

2 Specifies *peer domain* scope.

3 Specifies *management domain* scope.

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

Standard Output

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

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

Standard Error

All trace messages are written to standard error.

Examples

1. To remove the resource with the Name c175n05 from resource class IBM.Host, enter:
   ```
   rmrsrc -s 'Name == "c175n05"' IBM.Host
   ```
2. To remove the resource linked with resource handle: ″0x4017 0x0001 0x00000000 0x0069684c 0x0d52332b3 0xf3f54b45″, enter:
   ```
   rmrsrc -r "0x4017 0x0001 0x00000000 0x0069684c 0x0d52332b3 0xf3f54b45"
   ```

Location

/usr/sbin/rsct/bin/rmrsrc Contains the **rmrsrc** command

Related Information

Commands: **lsrsrc**, **mkrsrc**

Files: **rmccli**

Books: see the *RSCT Guide and Reference* for information about RMC operations

---

rmsensor Command

Purpose

Removes an event sensor command from the resource monitoring and control (RMC) subsystem.

Syntax

rmsensor [**–a** | **–n** *host1*[**,***host2*...]] [**–v** | **–V**] [**–h**] *name1* [*name2*...]

Description

The **rmsensor** command removes the sensor identified by *name* from RMC. This command returns the exit status and error messages of the underlying **rmrsrc** RMC command.

The **rmsensor** command can be run on the management server or on any node and normally operates on the local node where the command is run unless the CT_CONTACT environment variable is set to prevent this. If you want **rmsensor** to run on all of the nodes in a domain, use the CT_MANAGEMENT_SCOPE environment variable. See the **Environment variables** section for more information.

See the **mksensor** command for a more complete explanation of RMC sensors.

Flags

–a Removes sensors that match the specified name on all nodes in the domain.

–n *host1***[,***host2***...]**
Specifies the node from which the sensor should be removed. By default, the sensor is removed from the local node. This flag is only appropriate in a management domain or a peer domain.

–v | **–V**
Writes the command's verbose messages to standard output. Use this flag to see the underlying RMC command that is being run. This can help to interpret error messages that are displayed.

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

Parameters

name1 **[***name2***...]**
Specifies one or more names of sensors to remove.

Security

The user needs write permission for the **IBM.Sensor** resource class in order to run **rmsensor**. Permissions are specified in the access control list (ACL) file on the contacted system. See the *RSCT Guide and Reference* for details on the ACL file and how to modify it.

Exit Status

0 The command has run successfully.

1 An incorrect combination of and parameters has been entered.

6 No resources were found.

If an error occurs in the RMC layer, the RMC return code will be returned as the exit status.

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_MANAGEMENT_SCOPE
Determines the management scope used for the command's RMC daemon session.

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.

Examples

1. To remove the sensor named **NumLog**, enter:

```
rmsensor NumLog
```

Location

/usr/sbin/rsct/bin/rmsensor Contains the **rmsensor** command

Related Information

Commands: **chsensor**, **lssensor, mksensor**, **rmrsrc**

Books: see the *RSCT Guide and Reference* for information about the ACL authorization file.

---

rmserver Command

Purpose

Removes a subserver definition from the Subserver Type object class.

Syntax

rmserver -t *Type*

Description

The **rmserver** command removes an existing subserver definition from the Subserver Type object class.

Flags

-t *Type* Specifies the subserver name that uniquely identifies the existing subserver to be removed. The **rmserver** command is unsuccessful if the *Type* name is not known in the Subserver Type object class.

Security

Auditing Events: If the auditing subsystem has been properly configured and is enabled, the **rmserver** command will generate the following audit record (event) every time the command is executed:

Event	Information
SRC_Delserver	Lists in an audit log the name of the subserver definition that was deleted.

See ″Setting Up Auditing″ in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices* for details about selecting and grouping audit events, and configuring audit event data collection.

Files

/etc/objrepos/SRCsubsvr Specifies the SRC Subserver Configuration object class.

Related Information

System Resource Controller Overview in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*.

System Resource Controller (SRC) Overview for Programmers in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

The **auditpr** command, **chserver** command, **mkserver** command, **startsrc** command, **stopsrc** command.

Auditing Overview in *AIX 5L Version 5.2 Security Guide*.

---

rmsock Command

Purpose

Removes a socket that does not have a file descriptor.

Syntax

rmsock *Address TypeofAddress*

Description

The **rmsock** command removes a socket that does not have a file descriptor. It accepts a socket, tcpcb, inpcb, ripcb, or rawcb address and converts it to a socket address. All opened files in every process are then checked to find a match to the socket. If a match is not found, an abort action is performed on that socket regardless of the existence of the socket **linger** option. The port number held by the socket is released. If a match is found, its file descriptor and status of the owner process are displayed to the user. The results are passed to **syslogd** and recorded in the **/var/adm/rmsock.log** file.

Examples

1. To remove a socket from its socket address, type:

   ```
   rmsock 70054edc socket
   ```

   You do not need to specify the type of the socket. It can be a tcpcb, udp, raw, or routing socket.
2. To remove a socket from its inpcb address, type:

   ```
   rmsock 70054edc inpcb
   ```
3. To remove a socket from its tcpcb address, type:

   ```
   rmsock 70054ecc tcpcb
   ```

Files

/usr/sbin	Directory where the **rmsock** command resides.
/var/adm/rmsock.log	Contains the **rmsock.log** file.

Related Information

The **syslogd** daemon.

---

rmss Command

Purpose

Simulates a system with various sizes of memory for performance testing of applications.

Syntax

rmss -c *MemSize*

rmss -r

rmss -p

rmss [**-d** *MemSize*] [**-f** *MemSize*] [**-n** *NumIterations*] [**-o** *OutputFile*] [**-s** *MemSize*] *Command*

Implementation Specifics

This command is valid only on the POWER-based platform.

Flags

-c *MemSize*	Changes the simulated memory size to the *MemSize* value, which is an integer or decimal fraction in units of megabytes. The *MemSize* variable must be between 4MB and the real memory size of the machine. There is no default for the **-c** flag. **Note:** It is difficult to change the simulated memory size to less than 8MB, because of the size of inherent system structures such as the kernel.
-d *MemSize*	Specifies the increment between memory sizes to be simulated. The *MemSize* value is an integer or decimal fraction in units of megabytes. If the **-d** flag is omitted, the increment will be 8MB.
-f *MemSize*	Specifies the final memory size. You should finish testing the simulated system by executing the command being tested at a simulated memory size given by the *MemSize* variable, which is an integer or decimal fraction in units of megabytes. The *MemSize* variable must be between 4MB and the real memory size of the machine. If the **-f** flag is omitted, the final memory size will be 8MB. **Note:** It is difficult to finish at a simulated memory size of less than 8MB because of the size of inherent system structures such as the kernel.
-n *NumIterations*	Specifies the number of times to run and measure the command, at each memory size. There is no default for the **-n** flag. If the **-n** flag is omitted, during **rmss** command initialization, the **rmss** command will determine how many iterations of the command being tested are necessary to accumulate a total run time of 10 seconds, and then run the command that many times at each memory size. **Note:** The **rmss** command always executes the command once at each memory size prior to the executions that are measured. This prepares the simulation for the actual test.
-o *OutputFile*	Specifies the file into which to write the **rmss** report. If the **-o** flag is omitted, then the **rmss** report is written to the file **rmss.out**. In addition, the **rmss** report is always written to standard output.
-p	Displays the current simulated memory size.
-r	Resets the simulated memory size to the real memory size of the machine.
-s *MemSize*	Specifies the starting memory size. Start by executing the command at a simulated memory size specified by the *MemSize* variable, which is an integer or decimal fraction in units of megabytes. The *MemSize* variable must be between 4MB and the real memory size of the machine. If the **-s** flag is omitted, the starting memory size will be the real memory size of the machine. **Note:** It is difficult to start at a simulated memory size of less than 8MB, because of the size of inherent system structures such as the kernel.
Command	Specifies the command to be run and measured at each memory size. The *Command* parameter may be an executable or shell script file, with or without command line arguments. There is no default command.

Security

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

Examples

1. To change the memory size to 13.5MB, enter:

   ```
   rmss -c 13.5
   ```

2. To print the current memory size, enter:

   ```
   rmss -p
   ```

3. To reset the memory size to the real memory size of the machine, enter:

   ```
   rmss -r
   ```

4. To investigate the performance of the command `cc -O foo.c` on memory sizes 32, 24, 16, and 8MB; run and measure the command once at each memory size; and then write the report to the `cc.rmss.out` file, enter:

   ```
   rmss -s 32 -f 8 -d 8 -n 1 -o cc.rmss.out cc -O foo.c
   ```

5. To investigate the performance of the sequence of commands in the `foo.sh` shell script file on memory sizes starting at the real memory size of the machine and ending at 8MB, by increments of 8MB; let the **rmss** command determine the number of iterations to run and measure the `foo.sh` at file each memory size; and then write the **rmss** report to the `rmss.out` file (with all defaults used in this invocation of the **rmss** command), enter the following:

   ```
   rmss foo.sh
   ```

6. To investigate the performance of the executable `bar` on memory sizes from 8MB to 16MB, by increments of 0.5MB; run and measure `bar` twice at each memory size; and write the report to the `bar.rmss.out` file, enter:

   ```
   rmss -s 8 -f 16 -d .5 -n 2 -o bar.rmss.out bar
   ```

7. When any combination of the **-s**, **-f**, **-d**, **-n**, and **-o** flags is used, the **rmss** command runs as a driver program, which executes a command multiple times over a range of memory sizes, and displays statistics describing the command's performance at each memory size.

   An example of the report printed out by the **rmss** command follows:

   ```
   Hostname:  xray.austin.ibm.com
   Real memory size:   48.00 Mb
   Time of day:  Wed Aug  8 13:07:33 1990
   Command:  cc -O foo.c
   Simulated memory size initialized to  24.00 Mb.
   Number of iterations per memory size = 1 warmup + 1 measured = 2.
   Memory size  Avg. Pageins  Avg. Response Time   Avg. Pagein Rate
   (megabytes)                      (sec.)           (pageins/sec.)
      ---------------------------------------------------------------
   24.00             0.0              113.7                0.0
   22.00             5.0              114.8                0.0
   20.00             0.0              113.7                0.0
   18.00             3.0              114.3                0.0
   16.00             0.0              114.6                0.0
   14.00             139.0            116.1                1.2
   12.00             816.0            126.9                6.4
   10.00             1246.0           135.7                9.2
   8.00              2218.0           162.9                13.6
   ```

   This report was generated by the following command:

   ```
   rmss -s 24 -f 8 -d 2 -n 1 cc -O foo.c
   ```

   The top part of the report gives general information, including the machine that the **rmss** command was running on, the real memory size of that machine, the time and date, and the command that was being measured. The next two lines give informational messages that describe the initialization of the **rmss** command. Here, the **rmss** command displays that it has initialized the simulated memory size to 24MB, which was the starting memory size given with the **-s** flag. Also, the **rmss** command prints out the number of iterations that the command will be run at each memory size. The command is to be run twice at each memory size: once to warmup, and once when its performance is measured. The number of iterations was specified by the **-n** flag.

The lower part of the report provides the following for each memory size the command was run at:

- The memory size, along with the average number of page-ins that occurred while the command was run
- The average response time of the command
- The average page-in rate that occurred when the command was run.

Note: The average page-ins and average page-in rate values include all page-ins that occurred while the command was run, not just those initiated by the command.

Related Information

The **filemon** command, and **svmon** command.

---

rmssys Command

Purpose

Removes a subsystem definition from the subsystem object class.

Syntax

rmssys -s *Subsystem*

Description

The **rmssys** command removes an existing subsystem definition from the subsystem object class. It also removes any subservers and notify method definitions that exist for the subsystem being removed.

Flags

-s *Subsystem*	Specifies the name that uniquely identifies the subsystem to be removed. The **rmssys** command is unsuccessful if the subsystem name is not known in the subsystem object class. The **rmssys** command removes any subserver definitions from the Subserver Type object class that are defined for this subsystem, as well as any notify method definitions from the Notify object class that are defined for this subsystem.

Security

Auditing Events: If the auditing subsystem has been properly configured and is enabled, the **rmssys** command will generate the following audit record (event) every time the command is executed:

Event	Information
SRC_Delssys	Lists in an audit log the name of the subsystem being removed.

See ″Setting Up Auditing″ in *AIX 5L Version 5.2 Security Guide* for details about selecting and grouping audit events, and configuring audit event data collection.

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, **mkssys** command, **lssrc** command, **chssys** command, **rmserver** command, **refresh** command, **startsrc** command, **stopsrc** command, **traceson** command, **tracesoff** command.

System Resource Controller Overview in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*.

System Resource Controller (SRC) Overview for Programmers in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

Auditing Overview in *AIX 5L Version 5.2 Security Guide*.

---

rmt Command

Purpose

Allows remote access to magnetic tape devices.

Syntax

rmt

Description

The **rmt** command allows remote access to magnetic tape devices. The remote dump and restore programs use the **rmt** command as a remote magnetic tape protocol module. The **rmt** command is normally started with a **rexec** or **rcmd** subroutine.

The **rmt** command accepts requests specific to the manipulation of magnetic tapes, performs the commands, and then responds with a status indication. All responses are in ASCII and in one of two forms. Successful commands receive responses of `Axxx`, where `xxx` is an ASCII representation of a decimal number. Unsuccessful commands receive responses of `Eyyy error-message`, where `yyy` is one of the possible error numbers described in the **errno.h** file and `error-message` is the corresponding error string as printed from a call to the **perror** subroutine. The protocol is comprised of the following subcommands.

Subcommands

O*DeviceMode*	Opens the device specified by the *Device* parameter using the mode indicated by the *Mode* parameter. The value of the *Device* parameter is a full path name, and that of the *Mode* parameter is an ASCII representation of a decimal number suitable for passing to the **open** subroutine. An open device is closed before a new open operation is performed.
C*Device*	Closes the open device. The device specified with the *Device* parameter is ignored.
L*WhenceOffset*	Performs an **lseek** operation using the specified parameters. The **lseek** subroutine returns the response value.
W*Count*	Writes data onto the open device. From the connection, the **rmt** command reads the number of bytes specified by the *Count* parameter, ending if a premature end-of-file is encountered. The **write** subroutine returns the response value.
R*Count*	Reads, from the open device, the number of bytes of data specified by the *Count* parameter. The **rmt** command then performs the requested read operation and responds with `Azzz`, where `zzz` is the number of bytes read if the operation was successful. The data read is then sent. Otherwise, an error in the standard format is returned.

I*OperationCount* Performs an **STIOCTOP** ioctl subroutine using the specified parameters. The parameters are interpreted as the ASCII representations of the decimal values to place in the `mt op` and `mt count` fields of the structure used in the ioctl subroutine. The return value is the value of the *Count* parameter when the operation is successful.

Any other subcommand causes the **rmt** command to exit.

> **Note:** For the **R** and **W** subcommands, if the *Count* parameter specifies more bytes than the connection can handle, the data will be truncated to a size that can be handled.

Exit Status

This command returns the following exit values:

0 Successful completion.
>0 An error occurred.

Files

/usr/sbin/rmt	Contains the **rmt** command.
/usr/include/sys/errno.h	Describes the possible error numbers.

Related Information

The **rdump** command, **rrestore** command.

The **rexec** subroutine.

The **rmt** special file.

---

rmtcpip Command

Purpose

Removes the TCP/IP configuration for a host machine.

Syntax

rmtcpip

Description

The **rmtcpip** command removes TCP/IP configuration on a host machine. The basic functions of this command is:

- Removes the network interface configurations
- Restores **/etc/rc.tcpip** to the initial installed state
- Restores **/etc/hosts** to the initial installed state
- Removes the **/etc/resolv.conf** file
- Removes the default and static routes
- Sets the hostname to localhost
- Sets the hostid to 127.0.0.1
- Resets configuration database to the initial installed state

Notes:

1. Any daemon which is commented out by default in **/etc/rc.tcpip**, but running at the time this command is issued, is stopped.
2. Your version of the **/etc/hosts** file is saved as **/etc/hosts.save** prior to the **/etc/hosts** file being restored to the originally installed state.
3. Your version of the **/etc/resolv.conf** file is saved as **/etc/resolv.conf.save** prior to the removal of the **/etc/resolv.conf** file.

Security
This command can only be run by root.

Related Information
The **hostent** command, **hostname** command, **hostid** command, **mktcpip** command.

The **/etc/resolv.conf** file format.

---

rmtun Command

Purpose
Deactivates operational tunnel(s) and optionally removes tunnel definition(s).

Syntax
rmtun -v 4|**6 -t** *tid_list* | **all** [**-d**]

Description
Use the **rmtun** command to deactivate an active tunnel(s) and optionally remove tunnel definition(s). It also will remove the auto-generated filter rules created for the tunnel by the **gentun** command when the tunnel definition is removed from the tunnel database.

Flags

all	Deactivates and optionally removes all the tunnel(s).
tid_list	The list of the tunnel(s) you want to deactivate. The tunnel IDs can be separated by ″,″ or ″-″. You can use ″-″ to specify a range of IDs. For example, 1,3,5-7 specified there are five tunnel IDs in the list, 1, 3, 5, 6 and 7.
-d	Specifies that the tunnels are to be removed from the tunnel database. This is an optional flag.
-t	The list of the tunnel(s) you want to deactivate. If **-d** is specified, all the tunnel definitions in the list will also be removed from the tunnel database.
-v	The IP version of the tunnel. For the IP version 4 tunnel, use the value of **4**. For the IP version 6 tunnel, use the value of **6**.

Related Information
The **chtun** command, **exptun** command, **gentun** command, **imptun** command, **lstun** command, and **mktun** command.

---

rmuser Command

Purpose
Removes a user account.

Syntax

rmuser [**-R** *load_module*] [**-p**] *Name*

Description

The **rmuser** command removes the user account identified by the *Name* parameter. This command removes a user's attributes without removing the user's home directory and files. The user name must already exist as a string of 8 bytes or less. If the **-p** flag is specified, the **rmuser** command also removes passwords and other user authentication information from the **/etc/security/passwd** file.

For users that are created with an alternate Identification and Authentication (I&A) mechanism, the **-R** flag with the appropriate load module must be used to remove that user. The load modules are defined in the **/usr/lib/security/methods.cfg** file.

Only the root user or users with UserAdmin authorization can remove administrative users. Administrative users are those users with **admin=true** set in the /**etc/security/user** file.

You can 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 rmuser** fast path to run this command.

Flags

-p	Removes user password information from the **/etc/security/passwd** file.
-R *load_module*	Specifies the loadable I&A module used to remove the user.

Security

Access Control: This command should grant execute (x) access only to the root user and members of the security group. This command should be installed as a program in the trusted computing base (TCB). The command should be owned by the root user with the **setuid** (SUID) bit set.

Files Accessed:

Mode	File
rw	**/etc/passwd**
rw	**/etc/security/passwd**
rw	**/etc/security/user**
rw	**/etc/security/user.roles**
rw	**/etc/security/limits**
rw	**/etc/security/environ**
rw	**/etc/security/audit/config**
rw	**/etc/group**
rw	**/etc/security/group**

Auditing Events:

Event	Information
USER_Remove	user

Examples

1. To remove the user `davis` account and its attributes from the local system, enter:

   ```
   rmuser davis
   ```

2. To remove the user `davis` account and all its attributes, including passwords and other user authentication information in the **/etc/security/passwd** file, type:

   ```
   rmuser -p davis
   ```

3. To remove the user `davis`, who was created with the LDAP load module, type:

   ```
   rmuser -R LDAP davis
   ```

Files

/usr/sbin/rmuser	Contains the **rmuser** command.
/etc/passwd	Contains the basic attributes of users.
/etc/security/passwd	Contains password information.
/etc/security/limits	Defines resource quotas and limits for each user.
/etc/security/user	Contains the extended attributes of users.
/etc/security/user.roles	Contains the administrative role attributes of users.
/etc/security/environ	Contains environment attributes of users.
/etc/security/audit/config	Contains audit configuration information.
/etc/group	Contains the basic attributes of groups.
/etc/security/group	Contains the extended attributes of groups.

Related Information

The **chfn** command, **chgrpmem** command, **chsh** command, **chgroup** command, **chuser** command, **lsgroup** command, **lsuser** command, **mkgroup** command, **mkuser** command, **passwd** command, **pwdadm** command, **rmgroup** 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 Standalone System Security in *AIX 5L Version 5.2 Security Guide*.

For more information about administrative roles, refer to Users, Groups, and Passwords in *AIX 5L Version 5.2 Security Guide*.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.2 Web-based System Manager Administration Guide*.

---

rmvfs Command

Purpose

Removes entries in the **/etc/vfs** file.

Syntax

rmvfs *VfsName*

Description

The **rmvfs** command removes a *VfsName* entry from the **/etc/vfs** file. The *VfsName* parameter is the name of a virtual file system. The **rmvfs** command takes one argument, the name of the virtual file system type to be removed from the file. If this *VfsName* entry exists, it is removed from the file.

Examples

To remove the `newvfs` entry, enter:

```
rmvfs newvfs
```

Files

/etc/vfs Contains descriptions of virtual file system types.

Related Information

The **chvfs** command, **crvfs** command, **lsvfs** command, **mount** command.

The File Systems in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices* explains file system types, management, structure, and maintenance.

---

rmvirprt Command

Purpose

Removes a virtual printer.

Syntax

rmvirprt -q *PrinterQueueName* **-d** *QueueDeviceName*

Description

The **rmvirprt** command removes the virtual printer assigned to the *PrinterQueueName* and *QueueDeviceName* variable value. The **rmvirprt** command also removes the System Management Interface Tool (SMIT) Object Database Manager (ODM) objects associated with the specified queue and queue device.

You can use the Printer Queues application in Web-based System Manager to change printer characteristics.

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

Note: When the command **rmvirprt** is run from the command line, it does not remove the queue or queue device, nor does it check for any jobs running or queued on the specified queue and queue device. However, if SMIT is used to run this command interactively, the corresponding queue, queue device, and, optionally, printer device, are removed along with the virtual printer, if there are no jobs running or queued.

Flags

-d *QueueDeviceName* Specifies the name of the queue device to which the virtual printer is assigned.
-q *PrinterQueueName* Specifies the name of the print queue to which the virtual printer is assigned.

Examples

To remove the attribute values for the `mypro` virtual printer associated with the `proq` print queue, type:

```
rmvirprt  -d mypro  -q proq
```

Files

/etc/qconfig Contains the configuration file.
/usr/sbin/rmvirprt Contains the **rmvirprt** command.
/var/spool/lpd/pio/@local/custom/* Contains the customized virtual printer attribute files.

/var/spool/lpd/pio/@local/ddi/* Contains the digested virtual printer attribute files.

Related Information

The **chvirprt** command, **lsvirprt** command,**mkvirprt** command, **smit** command.

Printer Overview for System Management in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

Spooler Overview for System Management in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.2 Web-based System Manager Administration Guide*.

---

rmyp Command

Purpose

Removes the configuration for NIS.

Syntax

/usr/sbin/rmyp { **-s** | **-c** }

Description

The **rmyp** command removes everything from the system that is used to make NIS work. For example, the **rmyp** command removes all of the NIS maps and all of the entries in the **/etc/rc.nfs** file for the NIS daemons.

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

```
smit rmyp
```

Flags

-s Removes the server configuration from the system.
-c Removes the client configuration from the system.

Related Information

The **chslave** command, **mkclient** command, **mkslave** command, **smit** command.

The **ypbind** daemon, **yppasswdd** daemon, **ypserv** daemon, **ypupdated** daemon.

Network File System (NFS) Overview for System Management in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.2 Web-based System Manager Administration Guide*.

System Management Interface Tool in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices*.

Network Information Service (NIS) in *AIX 5L Version 5.2 Network Information Services (NIS and NIS+) Guide*.

NIS Reference.

---

rndc Command

Purpose

Name server control utility.

Syntax

rndc [**-c** *config-file*] [**-k** *key-file*] [**-s** *server*] [**-p** *port*] [**-V**] [**-y** *key_id*] [*command*]

Description

rndc controls the operation of a name server. It supersedes the **ndc** utility that was provided in old BIND releases. If **rndc** is invoked with no command line options or arguments, it prints a short summary of the supported commands and the available options and their arguments.

rndc communicates with the name server over a TCP connection, sending commands authenticated with digital signatures. In the current versions of **rndc** and **named**, the only supported authentication algorithm is HMAC-MD5, which uses a shared secret on each end of the connection. This provides TSIG-style authentication for the command request and the name server's response. All commands sent over the channel must be signed by a **key_id** known to the server.

rndc reads a configuration file to determine how to contact the name server and decide what algorithm and key it should use.

Flags

-c *config-file*	Use *config-file* as the configuration file instead of the default, **/etc/rndc.conf**.
-k *key-file*	Use ***key-file*** as the key file instead of the default, **/etc/rndc.key**. The key in **/etc/rndc.key** will be used to authenticate commands sent to the server if the *config-file* does not exist.
-s *server*	***server*** is the name or address of the server which matches a ***server*** statement in the configuration file for **rndc**. If no server is supplied on the command line, the host named by the default-server clause in the option statement of the configuration file will be used.
-p *port*	Send commands to TCP port port instead of BIND 9's default control channel port, 953.
-V	Enable verbose logging.
-y *keyid*	Use the key ***keyid*** from the configuration file. ***keyid*** must be known by named with the same algorithm and secret string in order for control message validation to succeed. If no ***keyid*** is specified, **rndc** will first look for a key clause in the server statement of the server being used, or if no server statement is present for that host, then the default-key clause of the options statement. Note that the configuration file contains shared secrets which are used to send authenticated control commands to name servers. It should therefore not have general read or write access.

For the complete set of commands supported by **rndc**, see the BIND 9 Administrator Reference Manual or run **rndc** without arguments to see its help message.

Limitations

The **rndc** command only works with the **named9** daemon. The shared-secret for a **key_id** cannot be provided without using the configuration file.

Related Information

The **rndc.conf(5)**, **named8**, **named.conf(5)** and **ndc(8)**, daemons.

The BIND 9 Administrator Reference Manual.

---

rndc-confgen Command

Purpose

Generates configuration files for the **rndc** command.

Syntax

rndc-confgen [**-a**] [**-b** *keysize*] [**-c** *keyfile*] [**-h**] [**-k** *keyname*] [**-p** *port*] [**-r** *randomfile*] [**-s** *address*]

Description

The **rndc-confgen** command generates configuration files for the **rndc** command. It can be used as a convenient alternative to writing the **rndc.conf** file and the corresponding controls and key statements in **named.conf** by hand. The **rndc-confgen** command can be run with the **-a** flag to set up a **rndc.key** file. Doing this will avoid the need for a **rndc.conf** file and a controls statement.

Note: The **rndc-confgen** command only works with the **named9** daemon.

Flags

-a	Performs automatic **rndc** configuration. This creates a file **rndc.key** in **/etc** (or whatever **sysconfdir** was specified as when BIND was built) that is read by both **rndc** and **named** on startup. The **rndc.key** file defines a default command channel and authentication key allowing **rndc** to communicate with **named** with no further configuration.
-b *keysize*	Specifies the size of the authentication key in bits. Must be between 1 and 512 bits. The default is 128.
-c *keyfile*	Used with the **-a** flag to specify an alternate location for **rndc.key**.
-h	Prints a short summary of the options and arguments to **rndc-confgen**.
-k *keyname*	Specifies the key name of the **rndc** authentication key. This must be a valid domain name. The default is **rndc-key**.
-p *port*	Specifies the command channel port where **named** listens for connections from **rndc**. The default is 953.
-r *randomfile*	Specifies a source of random data for generating the authoriazation. If the operating system does not provide a **/dev/random** or equivalent device, the default source of randomness is keyboard input. **randomdev** specifies the name of a character device or file containing random data to be used instead of the default. The special value keyboard indicates that keyboard input should be used.
-s *address*	Specifies the IP address where **named** listens for command channel connections from **rndc**. The default is the loopback address 127.0.0.1.

Examples

1. To allow **rndc** to be used with no manual configuration, run the following:

   ```
   rndc-confgen -a
   ```

2. To print a sample **rndc.conf** file and corresponding controls and key statements to be manually inserted into **named.conf**, run the following:

   ```
   rndc-confgen
   ```

Related Information

"rndc Command" on page 622 and "named Daemon" on page 1.

roffbib Command

Purpose
Prints a bibliographic database.

Syntax
roffbib [**-m** *Macro*] [**-x**] [*FormatFlags*] [*Database*...]

Description
The **roffbib** command prints out all records that are in a bibliographic database format rather than in a format for footnotes or endnotes. Generally, the command is used as a filter for the **sortbib** command.

The **roffbib** command accepts most of the flags used by the **nroff** command and the **troff** command, in particular, the **-e**, **-h**, **-n**, **-o**, **-r**, **-s**, and **-T** flags.

If abstracts or comments are entered following the **%X** key field, they are formatted into paragraphs for an annotated bibliography. Several **%X** fields can be given if several annotation paragraphs are desired.

Parameters

FormatFlags	Accepts most of the **nroff** command flags, especially the **-e**, **-h**, **-n**, **-o**, **-r**, **-s**, and **-T** flags.
Database	Stores a bibliographic database of all records.

Flags

-m *Macro*	Specifies a file that contains a user-defined set of macros. There should be a space between the **-m** flag and the macro. This set of macros replaces the ones defined in the **/usr/share/lib/tmac/tmac.bib** file. Users can rewrite macros to create customized formats.
-x	Suppresses the printing of abstracts or comments that are entered following the **%X** field key.

Examples
Following is an example of the **roffbib** command used in conjunction with the **sortbib** command.

```
sortbib Database | roffbib
```

Files

/usr/share/lib/tmac/tmac.bib	Contains macros used by the **nroff** and **troff** commands.

Related Information
The **addbib** command, **indxbib** command, **lookbib** command, **nroff** command, **refer** command, **sortbib** command, **troff** command.

route Command

Purpose
Manually manipulates the routing tables.

Syntax

route [**-f**] [**-n**] [**-q**] [**-v**] *Command* [*Family*] [[**-net** | **-host**] *Destination* [**-prefixlen** *n*] [**-netmask** [*Address*]] *Gateway*] [*Arguments*]

Description

The **route** command allows you to make manual entries into the network routing tables. The **route** command distinguishes between routes to hosts and routes to networks by interpreting the network address of the *Destination* variable, which can be specified either by symbolic name or numeric address. The **route** command resolves all symbolic names into addresses, using either the **/etc/hosts** file or the network name server.

Routes to a particular host are distinguished from those to a network by interpreting the Internet address associated with the destination. The optional keywords **-net** and **-host** force the destination to be interpreted as a network or a host, respectively. If the destination has a local address part of INADDR_ANY or if the destination is the symbolic name of a network, then the route is assumed to be to a network; otherwise, it is presumed to be a route to a host.

For example, `128.32` is interpreted as `-host 128.0.0.32`; `128.32.130` is interpreted as `-host 128.32.0.130`; `-net 128.32` is interpreted as `128.32.0.0`; and `-net 128.32.130` is interpreted as `128.32.130.0`.

If the route is by way of an interface rather than through a gateway, the **-interface** argument should be specified. The specified gateway is the address of the host on the common network, indicating the interface to be used for transmission.

The **-netmask** argument must be followed by an address parameter (to be interpreted as a network mask). One can override the implicit network mask generated in the **-inet** case by making sure this option follows the *Destination* parameter.

All symbolic names specified for a destination or gateway are looked up first as a host name, using the **gethostbyname** subroutine. If this fails, the **getnetbyname** subroutine is then used to interpret the name as a network name.

> **Note:** Route uses a routing socket and the new message types RTM_ADD, RTM_DELETE, and RTM_CHANGE. As such, only the root user may modify the routing tables.

If the **flush** or **-f** command is specified, route will ″flush,″ or clear, the routing tables of all gateway entries. One can choose to flush only those routes whose destinations are of a given address family, by specifying an optional keyword describing which address family.

The **netstat -r** command displays the current routing information contained in the routing tables.

Flags

-f	Purges all entries in the routing table that are not associated with network interfaces.
-n	Displays host and network names numerically, rather than symbolically, when reporting results of a flush or of any action in verbose mode.
-q	Specifies quiet mode and suppresses all output.
-v	Specifies verbose mode and prints additional details.
-net	Indicates that the *Destination* parameter should be interpreted as a network.
-netmask	Specifies the network mask to the destination address. Make sure this option follows the *Destination* parameter.
-host	Indicates that the *Destination* parameter should be interpreted as a host.
-prefixlen *n*	Specifies the length of a destination prefix (the number of bits in the netmask).

The route default is a host (a single computer on the network). When neither the **-net** parameter nor the **-host** parameter is specified, but the network portion of the address is specified, the route is assumed to be to a network. The host portion of the address is 0 (zero).

Parameters

Arguments Specifies one or more of the following arguments. Where *n* is specified as a variable to an argument, the value of the *n* variable is a positive integer.

-active_dgd
Enables Active Dead Gateway Detection on the route.

-cloning
Clones a new route.

-genmask
Extracts the length of TSEL, which is used for the generation of cloned routes.

-interface
Manipulates interface routing entries.

-rtt *n* Specifies round-trip time.

-rttvar *n*
Specifies round-trip time variance.

-sendpipe *n*
Specifies send-window size.

-recvpipe *n*
Specifies receive-window size.

-allowgroup *gid*
Specifies a group ID that is allowed to use the route. The group ID will be added to a list of allowed groups or deleted from a list of denied groups. This argument only applies to AIX 4.2.1 or later.

-denygroup *gid*
Specifies a group ID that is not allowed to use the route. The group ID will be added to a list of denied groups or deleted from a list of allowed groups. This argument only applies to AIX 4.2.1 or later.

-stopsearch
Stops searching if a routing table lookup matches the route, but it is not allowed to use the route due to group routing restrictions.

-mtu *n* Specifies maximum transmission unit for this route. Will override interface mtu for TCP applications as long as it does not exceed maximum mtu for the interface. This flag has no affect on mtu for applications using UDP.

-hopcount *n*
Specifies maximum number of gateways in the route.

-expire *n*
Specifies expiration metrics used by routing protocol

-ssthresh *n*
Specifies outbound gateway buffer limit.

-lock Specifies a meta-modifier that can individually lock a metric modifier. The **-lock** meta-modifier must precede each modifier to be locked.

-lockrest
Specifies a meta-modifier that can lock all subsequent metrics.

-if *ifname*
Specifies the interface (en0, tr0 ...) to associate with this route so that packets will be sent using this interface when this route is chosen.

-xresolve
Emits a message on use (for external lookup).

-iface Specifies that the destination is directly reachable.

-static Specifies the manually added route.

-nostatic
Specifies the pretend route that is added by the kernel or daemon.

-reject Emits an ICMP unreachable when matched.

-blackhole
Silently discards packets during updates.

-proto1
Sets protocol specific routing flag number 1.

-proto2
Sets protocol specific routing flag number 2.

Command Specifies one of six possibilities:

add Adds a route.

flush or -f
Removes all routes.

delete Deletes a specific route.

change
Changes aspects of a route (such as its gateway).

monitor
Reports any changes to the routing information base, routing lookup misses, or suspected network partitionings.

get Lookup and display the route for a destination.

Family Specifies the address family. The **-inet** address family is the default. The **-inet6** family specifies that all subsequent addresses are in the inet6 family, and the **-xns** address family specifies that all subsequent addresses are in the Xerox Network System (XNS) address family.

Destination Identifies the host or network to which you are directing the route. The *Destination* parameter can be specified either by symbolic name or numeric address.

Gateway Identifies the gateway to which packets are addressed. The *Gateway* parameter can be specified either by symbolic name or numeric address.

Examples

The following are examples using the **route** command on a TCP/IP network and an XNS network, respectively:

Inet Examples

1. To establish a route so that a computer on one network can send a message to a computer on a different network, enter:

   ```
   route add 192.100.201.7 192.100.13.7
   ```

   The `192.100.201.7` address is that of the receiving computer (the *Destination* parameter). The `192.100.13.7` address is that of the routing computer (the *Gateway* parameter).
2. To establish a route so you can send a message to any user on a specific network, enter:

   ```
   route add -net 192.100.201.0 192.100.13.7
   ```

   The `192.100.201.0` address is that of the receiving network (the *Destination* parameter). The `192.100.13.7` address is that of the routing network (the *Gateway* parameter).
3. To establish a default gateway, enter:

   ```
   route add 0 192.100.13.7
   ```

   The value `0` or the default keyword for the *Destination* parameter means that any packets sent to destinations not previously defined and not on a directly connected network go through the default gateway. The `192.100.13.7` address is that of the gateway chosen to be the default.
4. To clear the host gateway table, enter:

   ```
   route -f
   ```

XNS Examples

1. To add a route to an XNS interface, enter:

   ```
   route add -xns 120 110:02.60.8c.2c.a4.98
   ```

   This adds an `xns` route to destination network `120` through gateway `110:02.60.8c.2c.a4.98`.
2. To flush all the gateways in an XNS interface, enter:

   ```
   route flush -xns
   ```

Related Information

The **netstat** command.

The **routed** daemon.

The **gethostbyname** subroutine, **getnetbyname** subroutine.

The **/etc/hosts** file format.

TCP/IP Routing, Gateways, TCP/IP Addressing in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

Xerox Network Systems (XNS) Overview for Programming in *AIX 5L Version 5.2 Communications Programming Concepts*.

---

routed Daemon

Purpose

Manages network routing tables.

Syntax

Note: Use SRC commands to control the **routed** daemon from the command line. Use the **rc.tcpip** file to start the daemon at each system startup.

/usr/sbin/routed [**-d**] [**-g**] [**-t**] { **-s** | **-q** } [*LogFile*]

Description

The **/usr/sbin/routed** daemon manages the network routing tables in the kernel. Unlike the **gated** daemon, which supports all TCP/IP gateway protocols, the **routed** daemon only implements the Routing Information Protocol (RIP). Do not use the **routed** daemon when Exterior Gateway Protocol (EGP), Simple Network Management Protocol (SNMP), or Defense Communications Network Local-Network Protocol (HELLO) routing is needed. Use the **gated** daemon for these protocols.

Note: Results are unpredictable when the **gated** and **routed** daemons run on the same host.

The **routed** daemon finds interfaces to directly connected hosts and networks that are configured into the system and marked as **up**. (Mark networks as **up** using the **ifconfig** command.) If multiple interfaces are present, the **routed** daemon assumes that the local host forwards packets between networks. The **routed** daemon transmits a RIP request packet on each interface, using a broadcast message if the interface supports it.

The **routed** daemon then listens for RIP routing requests and response packets from other hosts. When the **routed** daemon supplies RIP information to other hosts, it sends RIP update packets every 30 seconds (containing copies of its routing tables) to all directly connected hosts and networks.

When the **routed** daemon receives a Routing Information Protocol (RIP) request packet to supply RIP routing information, the **routed** daemon generates a reply in the form of a response packet. The response packet is based on the information maintained in the kernel routing tables and contains a list of known routes. Each route is marked with a hop-count metric, which is the number of gateway hops between the source network and the destination network. The metric for each route is relative to the sending host. A metric of 16 or greater is considered infinite or beyond reach.

The **routed** daemon writes information about its actions to a *LogFile*. This log file contains information about any changes to the routing tables. The file also contains a history of recent route-change messages both sent and received.

Updating Routing Tables

The **routed** daemon uses information contained in RIP response and update packets from other hosts to update its routing tables when at least one of the following conditions exists:

- No routing table entry exists for the destination network or host, and the metric associated with the route is finite; that is, the metric is less than 16.
- The source host of the packet is the router in the existing routing table entry. That is, updated information is being received from the same internetwork router through which packets for the destination are being routed.
- The existing entry in the routing table has not been updated in the last 90 seconds, and the route is at least as efficient as the current route.
- The new route is shorter than the one to the same destination currently stored in the routing tables. The **routed** daemon determines relative route length by comparing the new metric with the one stored in the routing table.

When the **routed** daemon updates its internal routing tables, it generates a RIP update packet to all directly connected hosts and networks. Before updating the kernel routing tables, the **routed** daemon pauses briefly to allow any unstable conditions to stabilize.

Besides processing incoming RIP packets, the **routed** daemon also checks the internal routing table entries periodically. The metric for any entry that has not been updated for three minutes is set to infinity and marked for deletion. The deletion is delayed for 60 seconds so that information about the route that is not valid can be distributed throughout the network. A host that acts as a RIP router supplies its routing tables to all directly connected hosts and networks every 30 seconds.

Using Gateways

Besides its ability to manage routes to directly connected hosts and networks, the **routed** daemon also uses distant and external gateways. These gateways cannot be identified by RIP queries, so the **routed** daemon reads the **/etc/gateways** file for information about these distant and external gateways.

The **/etc/gateways** file contains information about routes through distant and external gateways to hosts and networks that should be advertised through RIP. These routes can be either static routes to specific destinations or default routes for use when a static route to a destination is unknown. The format of the **/etc/gateways** file is:

{ *net* | *host* } *name1 gateway name2 metric* { **passive** | **active** | **external** }

When a gateway specified in the **/etc/gateways** file supplies RIP routing information, it should be marked as active. Active gateways are treated like network interfaces. That is, RIP routing information is distributed to the active gateway. If no RIP routing information is received from the gateway for a period of time, the **routed** daemon deletes the associated route from the routing tables.

A gateway that does not exchange RIP routing information should be marked as passive. Passive gateways are maintained in the routing tables indefinitely. Information about passive gateways is included in any RIP routing information transmitted.

An external gateway is identified to inform the **routed** daemon that another routing process will install such a route and that the **routed** daemon should not install alternative routes to that destination. External gateways are not maintained in the routing tables and information about them is not included in any RIP routing information transmitted.

Note: Routes through external gateways must be to networks only.

The **routed** daemon can also perform name resolution when routing to different networks. For example, the following command adds a route to the network called `netname` through the gateway called `host1`. The `host1` gateway is one hop count away.

```
route add net netname host1 1
```

To perform network name resolution, the **routed** daemon uses the **/etc/networks** file to get information on the network addresses and their corresponding names. To perform host name resolution, the **routed** daemon must take additional steps before the routing is complete. First the daemon checks for the existence of the **/etc/resolv.conf** file. This file indicates whether the host is running under a domain name server, and if so, gives the IP address of the host machine running the **named** daemon.

If the **/etc/resolv.conf** file does not exist, the **routed** daemon uses the **/etc/hosts** file to find the host for which it is routing.

The **routed** daemon should be controlled using the System Resource Controller (SRC) or the System Management Interface Tool (SMIT). Entering the **routed** daemon at the command line is not recommended.

Manipulating the routed Daemon with the System Resource Controller

The **routed** daemon is a subsystem controlled by the System Resource Controller (SRC). The **routed** daemon is a member of the SRC **tcpip** system group. This daemon is disabled by default and can be manipulated by the following SRC commands:

startsrc	Starts a subsystem, group of subsystems, or subserver.
stopsrc	Stops a subsystem, group of subsystems, or subserver.
traceson	Enables tracing of a subsystem, group of subsystems, or subserver.
tracesoff	Disables tracing of a subsystem, group of subsystems, or subserver.
lssrc	Gets the status of a subsystem, group of subsystems, or subserver.

Signals

The following signals have the specified effect when sent to the **routed** process using the **kill** command:

SIGINT	Restarts the **routed** daemon and flushes the routing table.
SIGHUP, **SIGTERM**, or **SIGQUIT**	Broadcasts RIP packets with hop counts set to infinity. These signals disable the local host as a router. After a second **SIGHUP**, **SIGTERM**, or **SIGQUIT** signal, the **routed** daemon terminates.
SIGUSR1	Turns packet tracing on or, if packet tracing is already on, steps up the tracing one level. The first level traces transactions only. The second level traces transactions plus packets. The third level traces the packet history, reporting packet changes. The fourth level traces packet contents. This command increments the level of tracing through four levels.
SIGUSR2	Turns packet tracing off.

Flags

-d Enables additional debugging information, such as bad packets received, to be logged.

-g Runs the routing daemon on a gateway host. The **-g** flag is used on internetwork routers to offer a route to the default destination.

-q Prevents the **routed** daemon from supplying routing information regardless of whether it is functioning as an internetwork router. The **-q** flag indicates ″quiet″. Do not use the **-q** flag and the **-s** flag together.

-s Supplies routing information regardless of whether it is functioning as an internetwork router. The **-s** flag indicates ″supply″. Do not use the **-q** flag and the **-s** flag together.

-t Writes all packets sent or received to standard output or to the file specified in the *LogFile* parameter. The **routed** daemon remains under control of the controlling terminal that started it. Therefore, an interrupt from the controlling terminal keyboard stops the **routed** process.

Examples

1. To start the **routed** daemon manually, type:

   ```
   startsrc -s routed -a "-s"
   ```

   Note: The **routed** daemon is not started by default at each system startup. Use the **rc.tcpip** file format and a System Resource Controller (SRC) command to start the **routed** daemon. You can also start the **routed** daemon using the System Management Interface Tool (SMIT).

   The **-s** flag causes the **routed** daemon to return routing information regardless of whether the **routed** daemon is an internetwork router.

2. To stop the **routed** daemon, type the following:

   ```
   stopsrc -s routed
   ```

3. To get a short-status report from the **routed** daemon, type the following:

```
lssrc -s routed
```

This command returns the name of the daemon, the process ID of the daemon, and the state of the daemon (active or inactive).

4. To enable tracing for **routed** daemon, type the following:

```
traceson -s routed
```

This command enables socket-level debugging. Use the **trpt** command to look at the output of this example command.

5. To trace and log all the packet routing information into the temporary file `logfile`, type the following:

```
startsrc -s routed -a "-t /tmp/logfile"
```

Related Information

The **route** command.

The **gated** daemon.

The **/etc/gateways** file format, **/etc/networks** file format.

TCP/IP Routing, TCP/IP Daemons, TCP/IP Protocols in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices*.

---

rpc.nisd Daemon

Purpose

Implements the NIS+ service.

Syntax

/usr/sbin/rpc.nisd [**-A**] [**-C**] [**-D**] [**-F**] [**-h**] [**-v**] [**-Y**] [**-c** *Seconds*] [**-d** *Dictionary*] [**-L** *Load*] [**-S** *Level*]

Description

The **rpc.nisd** daemon is a remote procedure call service that implements the NIS+ service. This daemon must be running on all servers that serve a portion of the NIS+ namespace. **rpc.nisd** is usually started from a system startup script.

Flags

-A	Sets the **rpc.nisd** daemon in authentication verbose mode. The daemon logs all the authentication-related activities to **syslogd** with **LOG_INFO** priority.
-C	Open diagnostic channel on **/dev/console**.
-D	Sets the **rpc.nisd** daemon in debug mode (doesn't fork).
-F	Forces the server to do a checkpoint of the database when it starts up. Forced checkpoints may be required when the server is low on disk space. The **-F** flag removes updates from the transaction log that have been propagated to all the replicas.
-h	Prints a list of options.
-v	Sets the **rpc.nisd** daemon in verbose mode. With the **-v** flag, the **rpc.nisd** daemon sends a running narration of its operations to the **syslog** daemon (see **syslog** at **LOG_INFO** priority). This flag is most useful for debugging problems with the NIS+ service (see also the **-A** flag).

-Y Sets the server in NIS (YP) compatibility mode. When operating in this mode, the NIS+ server responds to NIS Version 2 requests using the Version 2 protocol. Because the YP protocol is not authenticated, only those items that do not have read access to anybody are visible through the Version 2 protocol. The Version 2 protocol supports only the standard Version 2 maps in this mode (see the **-B** flag).

-c *Seconds* Sets the number of seconds between pushing out for updates to the server's replicas. The default is 120 seconds (two minutes).

-d *Dictionary* Specifies an alternate dictionary for the NIS+ database. The primary use of the **-d** flag is for testing. Note that the string is not interpreted; instead, it is passed on to the **db_initialize** function.

-L *Load* Specifies the maximum number of child processes that the server may spawn. The value of *Load* must be at least 1 for the callback functions to work correctly. The default is 128.

-S *Level* Sets the authorization security level of the **rpc.nisd** daemon. The value of the *Level* parameter must be between 0 and 2. The default is 2. The following values indicate these security levels:

0 At security level 0 the **rpc.nisd** daemon does not enforce any access controls. Any client is allowed to perform any operation, including updates and deletions. The 0 security level is intended for testing and initial setup of the NIS+ namespace.

1 At security level 1 the **rpc.nisd** daemon accepts both **AUTH_SYS** and **AUTH_DES** credentials for authenticating and authorizing clients to perform NIS+ operations. Level 1 is not a secure mode of operation because **AUTH_SYS** credentials are easy to forge. You should not use this security level on networks where any unknown user might have access.

2 At security level 2 the **rpc.nisd** daemon accepts only **AUTH_DES** credentials for authentication and authorization. 2 is the highest level of security provided by the NIS+ service and the default.

Environment

NETPATH Limits the transports available for NIS+ to use.

Examples

1. To set up the NIS+ service, enter:
   ```
   rpc.nisd
   ```
2. To set the NIS+ service in YP compatibility mode with DNS forwarding, enter:
   ```
   rpc.nisd -YB
   ```

Files

/var/nis/parent.object Contains an XDR-encoded NIS+ object describing the namespace above a root server. This parent namespace can be another NIS+ namespace or a foreign namespace such as the one served by the Domain Name Server. The **/var/nis/parent.object** only exists on servers serving the root domain namespace.

/var/nis/root.object Contains an XDR-encoded NIS+ object that describing the root of the namespace. The **/var/nis/root.object** file only exists on servers serving the root of the namespace.

/etc/init.d/rpc Contains the initialization script for NIS+.

Related Information

The **nis_cachemgr** daemon, the **nisinit** command, the **nissetup** command, and the **rpc.nispasswdd** daemon.

---

rpc.nispasswdd Daemon

Purpose

NIS+ password update daemon.

Syntax

/usr/sbin/rpc.nispasswdd [[**-a** *Attempts*] [**-c** *Minutes*] [**-D**] [**-g**] [**-v**]

Description

The **rpc.nispasswdd** daemon is an ONC+ RPC service that services password update requests from **nispasswd** and **yppasswd**. It updates password entries in the NIS+ **passwd** table.

The **rpc.nispasswdd** daemon is normally started from a system startup script after the NIS+ server, **rpc.nisd** has been started. **rpc.nispasswdd** determines whether it is running on a machine that is a master server for one or more NIS+ directories. If it discovers that the host is not a master server, then it promptly exits. It also determines if **rpc.nisd** is running in NIS(YP) compatibility mode (the **-Y** flag and registers as **yppasswdd** for NIS(YP) clients as well.

The **rpc.nispasswdd** daemon will syslog all failed password update attempts, which allows an administrator to determine whether someone was trying to ″crack″ the passwords.

rpc.nispasswdd has to be run by a superuser.

Flags

-a *Attempts*	Sets the maximum number of attempts allowed to authenticate the caller within a password update request session. Failed attempts are **syslogd** and the request is cached by the daemon. After the maximum number of allowed attempts the daemon severs the connection to the client. The default value is set to 3.
-c *Minutes*	Sets the number of minutes a failed password update request should be cached by the daemon. This is the time during which if the daemon receives further password update requests for the same user and authentication of the caller fails, then the daemon will simply not respond. The default value is set to 30 minutes.
-D	Runs in debugging mode.
-g	Generates DES credential. By default the DES credential is not generated for the user if they do not have one. By specifying this flag, if the user does not have a credential, then one will be generated for them and stored in the NIS+ cred table.
-v	Sets verbose mode. With this flag, the daemon sends a running narration of what it is doing to the syslog daemon. This flag is useful for debugging problems.

Exit Status

0	Success
1	An error has occurred.

Files

/etc/init.d/rpc	Initialization script for NIS+

Related Information

The **passwd** command, **yppasswd** command.

The **rpc.nisd** daemon, **syslogd** daemon.

---

rpc.pcnfsd Daemon

Purpose

Handles service requests from PC-NFS (Personal Computers Network File System) clients.

Syntax

/usr/sbin/rpc.pcnfsd

Description

The **rpc.pcnfsd** daemon handles requests from PC-NFS clients for authentication services on remote machines. These services include authentication for mounting and for print spooling. The PC-NFS program allows personal computers running DOS to be networked with machines running NFS. The **rpc.pcnfsd** daemon supports Versions 1 and 2 of the **pcnfsd** protocol.

When a PC-NFS client makes a request, the **inetd** daemon starts the **rpc.pcnfsd** daemon (if the **inetd.conf** file contains the appropriate entry). The **rpc.pcnfsd** daemon reads the **/etc/pcnfsd.conf** configuration file, if present, then services RPC requests directed to program number 150001. After the **rpc.pcnfsd** daemon is started, all print requests go to the default print spooling directory (**/var/spool/pcnfs**).

Authentication

When it receives a **PCNFSD_AUTH** or **PCNFSD2_AUTH** request, the **rpc.pcnfsd** daemon logs in a user by first validating the user name and password, and then returning the corresponding user ID (UID), group ID (GIDS), home directory, and **umask** specifications. A record of logins is appended to the **/var/adm/wtmp** file. To disable the login record feature, add the following line to the **/etc/pcnfsd.conf** file:

```
wtmp off
```

Printing

The **rpc.pcnfsd** daemon uses the Network File System (NFS) to transfer print data between clients and servers. The client system first issues a **PCNFSD_PR_INIT** or **PCNFSD2_PR_INIT** request. The server then returns a spool directory path for the client to use.

Note: The spool directory must be exported by NFS, using the **exportfs** command and the **/etc/exports** file.

The **rpc.pcnfsd** daemon creates a subdirectory for each of its clients. The default parent directory is **/var/spool/pcnfs**, and the subdirectory is the host name of the client system. To use a different parent directory, add the following line to the **/etc/pcnfsd.conf** file:

```
spooldir Pathname
```

When the spool directory is mounted and the print data is transferred to a file in this directory, the client issues a **PCNFSD_PR_START** or **PCNFSD2_PR_START** request. The **rpc.pcnfsd** daemon handles this (and most other print-related requests) by constructing an **enq** command. The daemon adopts the identity of the personal computer user to execute the print request command. Because constructing and executing the command involves user ID privileges, the **rpc.pcnfsd** daemon must be run as a root process.

All print requests from clients include the name of the printer to be used. The printer name is represented by queue and device definitions in the **/etc/qconfig** file. Additionally, the **rpc.pcnfsd** daemon provides a method for defining PC-NFS virtual printers recognized only by **rpc.pcnfsd** clients. Each PC-NFS virtual printer is defined in the **/etc/pcnfsd.conf** file with a line similar to the following:

```
printer Name AliasFor Command
```

In this format, `Name` specifies the name of the printer to be defined, and `AliasFor` is the name of the existing printer that will do the work. For example, a request to show the queue for `Name` translates into a queue command on the `AliasFor` printer. To define a printer `Name` with no existing printer, use a single - (minus sign) in place of the `AliasFor` parameter. The `Command` parameter specifies a command run when a file is printed on the `Name` printer. This command is executed by the Bourne shell, using the **-c** option. For complex operations, replace the `Command` parameter with an executable shell script.

The following list of tokens and substitution values can be used in the *Command* parameter:

Token	Substitution Value
$FILE	The full path name of the print data file. After the command has executed, the file is unlinked.
$USER	The user name of the user logged-in to the client.
$HOST	The host name of the client system.

Examples

The following example **/etc/pcnfsd.conf** file configures a virtual printer on the first line and a null device for testing on the second line:

```
printer rotated lw /bin/enscript -2r $FILE
printer test - /usr/bin/cp $FILE /usr/tmp/$HOST-$USER
```

The first line stipulates that if a client system prints a job on the `rotated` printer, the `enscript` utility is called to preprocess the `$FILE` file. The `-2r` option causes the file to be printed in two-column, rotated format on the default PostScript printer. If a client requests a list of the print queue for the `rotated` printer, the **rpc.pcnfsd** daemon translates this request into a request for a similar listing for the `lw` printer.

The second line establishes a printer test. Files sent to the `test` printer are copied into the **/usr/tmp** directory. Requests to the `test` printer to list the queue, check the status, or perform similar printer operations, are rejected because - (minus sign) is specified in place of the *AliasFor* parameter.

Files

/etc/inetd.conf	Contains the TCP/IP configuration file that starts RPC daemons and other TCP/IP daemons.
/etc/pcnfsd.conf	Contains the **rpc.pcnfsd** daemon configuration file.
/var/spool/pcnfs	Contains the default print-spooling directory.

Related Information

The **enq** command, **last** command.

The **inetd** daemon.

The **umask** subroutine.

Network File System (NFS) Overview for System Management in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

Printer Overview for System Management in the *AIX 5L Version 5.2 Guide to Printers and Printing*.

List of NFS Commands in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

rpcgen Command

Purpose

Generates C code to implement an RPC protocol.

Syntax

To Generate Four Types of Output Files for a File

/usr/bin/rpcgen *InputFile*

To Generate a Specific Output File for a File

rpcgen { **-c** | **-h** | **-l** | **-m** } [**-o** *OutputFile*] [*InputFile*]

To Generate a Server-Side File for TCP or UDP

rpcgen { **-s** *Transport ...* } [**-o** *OutputFile*] [*InputFile*]

Description

The **rpcgen** command generates C code to implement a Remote Procedure Call (RPC) protocol. The input to the **rpcgen** command is a language similar to C language known as RPC Language.

The first syntax structure is the most commonly used form for the **rpcgen** command where it takes an input file and generates four output files. For example, if the *InputFile* parameter is named **proto.x**, then the **rpcgen** command generates the following:

proto.h	Header file
proto_xdr.c	XDR routines
proto_svc.c	Server-side stubs
proto_clnt.c	Client-side stubs

Use the other syntax structures when you want to generate a particular output file rather than all four output files.

The **cpp** command, a C preprocessor, is run on all input files before they are actually interpreted by the **rpcgen** command. Therefore, all the **cpp** directives are legal within an **rpcgen** input file. For each type of output file, the **rpcgen** command defines a special **cpp** symbol for use by the **rpcgen** programmer:

RPC_HDR	Defined when compiling into header files
RPC_XDR	Defined when compiling into XDR routines
RPC_SVC	Defined when compiling into server-side stubs
RPC_CLNT	Defined when compiling into client-side stubs

In addition, the **rpcgen** command does some preprocessing of its own. Any line beginning with a **%** (percent sign) passes directly into the output file, uninterpreted by the **rpcgen** command.

To create your own XDR routines, leave the data types undefined. For every data type that is undefined, the **rpcgen** command assumes that a routine exists by prepending **xdr**_ to the name of the undefined type.

Notes:

1. Nesting is not supported. As a work-around, structures can be declared at top-level with their names used inside other structures in order to achieve the same effect.
2. Name clashes can occur when using program definitions since the apparent scoping does not really apply. Most of these can be avoided by giving unique names for programs, versions, procedures, and types.
3. To program to the TIRPC interfaces, and allow the use of multi-threaded RPC applications use the **tirpcgen** command. It will also be necessary to define the preprocessor variable **_AIX_TIRPC** in the Makefile as well as the **libtli.a** (**-ltli**) specification. **tirpcgen** is a temporary name for a new **rpcgen** command that will replace **rpcgen** in a future version the operating system.

Flags

-c	Compiles into XDR routines.
-h	Compiles into C-data definitions (a header file).
-l	Compiles into client-side stubs.
-m	Compiles into server-side stubs, but does not generate a main routine. This option is useful for doing call-back routines and for writing a main routine to do initialization.
-o *OutputFile*	Specifies the name of the output file. If none is specified, standard output is used.
-s *Transport*	Compiles into server-side stubs, using given transport. The supported transports are `udp` and `tcp`. This flag can be run more than once to compile a server that serves multiple transports.

Related Information

The **cpp** command.

Network File System (NFS) Overview for System Management in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

Remote Procedure Call (RPC) Overview for Programming in *AIX 5L Version 5.2 Communications Programming Concepts*.

List of NFS Commands.

---

rpcinfo Command

Purpose

Reports the status of Remote Procedure Call (RPC) servers.

Syntax

To Display a List of Statistics

/usr/bin/rpcinfo [**-m** | **-s**] [*Host*]

To Display a List of Registered RPC Programs

/usr/bin/rpcinfo -p [*Host*]

To Report Transport

/usr/bin/rpcinfo -T *transport Host Prognum* [*Versnum*]

To Display a List of Entries

/usr/bin/rpcinfo -l [**-T** *transport*] *Host Prognum Versnum*

To Report Program Status

/usr/bin/rpcinfo [**-n** *PortNum*] **-u** *Host Prognum* [*Versnum*]

To Report Response Status
/usr/bin/rpcinfo [**-n** *PortNum*] **-t** *Host Prognum* [*Versnum*]

To Display All Hosts Running a Specified Program Version
/usr/bin/rpcinfo [**-b**] [**-T** *transport*] *Prognum Versnum*

To Delete Registration of a Service
/usr/bin/rpcinfo [**-d**] [**-T** *transport*] *Prognum Versnum*

Description
The **rpcinfo** command makes an RPC call to an RPC server and reports the status of the server. For instance, this command reports whether the server is ready and waiting or not available.

The program parameter can be either a name or a number. If you specify a version, the **rpcinfo** command attempts to call that version of the specified program. Otherwise, the **rpcinfo** command attempts to find all the registered version numbers for the program you specify by calling version 0 (zero) and then attempts to call each registered version. (Version 0 is presumed not to exist. If it does exist, the **rpcinfo** command attempts to obtain this information by calling an extremely high version number instead.)

Flags

-b	Makes an RPC broadcast to procedure 0 of the specified prognum and versnum and reports all hosts that respond. If *transport* is specified, it broadcasts its request only on the specified *transpor*t. If broadcasting is not supported by any *transport*, an error message is printed. Using broadcasting (**-b** flag) should be limited because of the possible adverse effect on other systems.
-d	Deletes registration for the RPC service of the specified prognum and versnum. If transport is used, unregister the service only on that transport, otherwise unregister the service on all the transports where it was registered. This option can be exercised only by the root user.
-l	Displays a list of entries with the specified prognum and versnum on the specified host. Entries are returned for all transports in the same protocol family as those used to contact the remote **portmap** daemon. This flag only applies to AIX 4.2.1 or later.
-m	Displays a table of portmap operations statistics on the specified host. The table contains statistics for each version of portmap (Versions 2, 3, and 4), the number of times each procedure was requested and successfully serviced, the number and type of remote call requests that were made, and information about RPC address lookups that were handled. This information is used for monitoring RPC activities on the host. This flag only applies to AIX 4.2.1 or later.
-n *Portnum*	Use the *Portnum* parameter as the port number for the **-t** and **-u** options instead of the port number given by the portmap. Using the **-n** options avoids a call to the remote portmap to find out the address of the service. This option is made obsolete by the **-a** option.
-p	Probes the **portmap** service on the host using Version 2 of the portmap protocol and displays a list of all registered RPC programs. If a host is not specified, it defaults to the local host.
-s	Displays a concise list of all registered RPC programs on the host. If host is not specified, the default is the local host. This flag only applies to AIX 4.2.1 or later.
-t	Makes an RPC call to procedure 0 of prognum on the specified host using TCP, and reports whether a response was received. This option is made obsolete when using the **-T** option as shown in the third syntax.
-T	Specifies the transport where the service is required. This flag only applies to AIX 4.2.1 or later.
-u	Makes an RPC call to procedure 0 of prognum on the specified host using UDP, and reports whether a response was received. This option is made obsolete when using the **-T** option as shown in the third syntax.

Examples
1. To show all of the RPC services registered on a local machine, enter:

   ```
   rpcinfo -p
   ```
2. To show all of the RPC services registered on a specific machine, enter:

```
rpcinfo  -p zelda
```

In this example, the **rpcinfo** command shows all RPC services registered on a machine named `zelda`.

3. To show all machines on the local network that are running a certain version of a specific server, enter:

```
rpcinfo  -b ypserv 2
```

In this example, the **rpcinfo** command shows a list of all machines that are running version `2` of the **ypserv** daemon.

4. To delete the registration of a service, enter:

```
rpcinfo  -d sprayd 1
```

In this example, the **rpcinfo** command deletes version `1` of the **sprayd** daemon.

Files

/etc/rpc
: Contains a list of server names and their corresponding RPC program numbers and aliases.

/etc/services
: Contains an entry for each service available through the Internet network.

Related Information

The **portmap** daemon.

Network File System (NFS) Overview for System Management in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

List of NFS Commands.

---

rrestore Command

Purpose

Copies previously backed up file systems from a remote machine's device to the local machine.

Syntax

rrestore [**-b***Number*] [**-h**] [**-i**] [**-m**] [**-s***Number*] [**-t**] [**-v**] [**-y**] [**-x**] [**-r**] [**-R**] **-f***Machine***:***Device* [*FileSystem ...*] [*File* ...]

Description

The **rrestore** command restores Version 3 by i-node backups from a remote machine's device to a file system on the local machine. The **rrestore** command creates a server on the remote machine to the backup medium.

The **rrestore** command only accepts backup formats created when a file system is backed up by i-node.

Note: A user must have root authority to execute this command.

Flags

-b*Number*
Specifies the number of blocks to read in a single input operation. If you do not specify this flag, the **rrestore** command selects a default value appropriate for the physical device you have selected. Larger values of the *Number* variable result in larger physical transfers from tape devices.

-f*Machine***:***Device*
Specifies the input device on the remote machine. Specify the *Device* variable as a file name (such as the **/dev/rmt0** file) to get input from the named device. For more information on using tape devices see the **rmt** special file.

-h
Restores only the actual directory named by the *File* parameter, not the files contained in that directory. This option is ignored when either the **-r** or **-R** flag is specified.

-i
Starts the interactive mode. This flag allows you to restore selected files from the directory represented by the *File* parameter. The subcommands for the **-i** flag are:

ls [*Directory***]**
Displays directory names within the specified *Directory* parameter with a / (slash) after the name, and displays files to be restored with an * (asterisk) before the name. If the **-v** flag is used, the i-node number of each file and directory is also displayed. If the *Directory1* parameter is not specified, the current directory is used.

cd *Directory*
Changes the current directory to the *Directory* parameter.

pwd
Displays the full path name of the current directory.

add [*File***]**
Specifies the *File* parameter to restore. If the *File* parameter is a directory, that directory and all its files are restored (unless the **-h** flag is used). Files to be restored are displayed with an * (asterisk) before the name by the **ls** subcommand. If the *File* parameter is not specified, the current directory is used.

delete [*File***]**
Specifies the *File* parameter to ignore in restore. If the *File* parameter is a directory, the directory and all its files are not restored (unless the **-h** flag is used). If the *File* parameter is not specified, the current directory is used.

extract
Restores all files displayed with an * (asterisk) before the name by the **ls** subcommand.

setmodes
Sets owner, modes, and times for the files being restored rather than using this information as it resides on the backup medium.

verbose
Displays the i-node numbers of all restored files with the **ls** subcommand. Information about each file is also displayed as it is restored. The next invocation of the **verbose** subcommand turns **verbose** off.

help
Displays a summary of the subcommands.

quit
Stops execution of the **rrestore** command immediately, even if all files requested have not been restored.

-m
Restores files by i-node number rather than by path name.

-r Restores an entire file system.

> **Attention:** If you do not follow this procedure carefully, you can ruin an entire file system. If you are restoring a full (level 0) backup, run the **mkfs** command to create an empty file system before doing the restore. To restore an incremental backup at level 2, for example, run the **mkfs** command, restore the appropriate level 0 backup, restore the level 1 backup, and finally restore the level 2 backup. As an added safety precaution, run the **fsck** command after you restore each backup level.

-R Causes the **rrestore** command to request a specific volume in a multivolume set of backup medium when restoring an entire file system. The **-R** flag provides the ability to interrupt and resume the **rrestore** command.

-s*Number* Specifies which backup to restore from a multibackup medium. Numbering starts with 1.

-t Displays the table of contents for the backed up files. The **rrestore** command displays the file name. The names are relative to the root (**/**) directory of the file system backed up. The only exception is the root (**/**) directory itself.

-v Reports the progress of the restoration as it proceeds.

-x Restores individually named files. If no names are given, all files on that medium are restored. The names must be in the same form as the names shown by the **-t** flag.

-y Prevents the **rrestore** command from asking whether it should stop the restore if a tape error is encountered. The **rrestore** command attempts to skip over bad blocks.

-? Displays the usage message.

Exit Status

This command returns the following exit values:

0 Successful completion.
>0 An error occurred.

Examples

1. To list files from a remote tape device, enter:

   ```
   rrestore  -fmachine1:/dev/rmt0  -t
   ```

   This command reads information from the `/dev/rmt0` device on remote `machine1`. The file names are shown.
2. To restore files, enter:

   ```
   rrestore  -x  -fmachine1:/dev/rmt0 /home/mike/file1
   ```

   This command extracts the `/home/mike/file1` file from the backup medium on the `/dev/rmt0` device on remote `machine1`.
3. To restore all the files in a directory, enter:

   ```
   rrestore -fhost:/dev/rmt0 -x /home/mike
   ```

   This command restores the directory `/home/mike` and all the files it contains.
4. To restore a directory, but not the files in the directory, enter:

   ```
   rrestore -fhost:/dev/rmt0 -x -h /home/mike
   ```
5. To restore all the files in a directory from a specific backup on a multibackup medium, enter:

   ```
   rrestore -s3 -fhost:/dev/rmt0.1 -x /home/mike
   ```

This command restores the /home/mike directory and all the files it contains from the third backup on the backup medium.

Files

/dev/rfd0	Specifies the default restore device.
/usr/sbin/rrestore	Contains the **rrestore** command.

Related Information

backup command, **rdump** command, **restore** command, **mkfs** command, **fsck** command, **dd** command.

filesystems file, **backup** file, **rmt** special file.

Backup Overview for System Management in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices*.

File Systems in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices*.

Directory Overview in *AIX 5L Version 5.2 Network Information Services (NIS and NIS+) Guide*.

Files Overview in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices*.

Rsh Command

Purpose

Invokes the restricted version of the Bourne shell.

Syntax

Rsh [**-i**] [{ **+** | **-** } { [**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 **Rsh** command invokes a restricted version of the Bourne shell, which is useful for installations that require a more controlled shell environment. The restricted shell allows you to create user environments with a limited set of privileges and capabilities.

For more information about the Bourne shell, see ″Bourne Shell″ in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices*.

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 *AIX 5L Version 5.2 System User's Guide: Operating System and Devices* 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. (Usually 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.

-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/sh	Specifies a default path name to the shell. This file is a link to the Bourne shell.
/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, **bsh** command.

The **/etc/passwd** file, **null** special file, **environment** file.

The **profile** file format.

Bourne Shell in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices*.

Bourne Shell Special Commands in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices*.

Variable Substitution in the Bourne Shell in *AIX 5L Version 5.2 System User's Guide: Operating System and Devices*.

---

rsh or remsh Command

Purpose

Executes the specified command at the remote host or logs into the remote host.

Syntax

{ **rsh** | **remsh** } *RemoteHost* [**-a**] [**-n**] [**-l** *User*] [**-f** | **-F**] [**-k** *realm*] [*Command*]

Description

The **/usr/bin/rsh** command executes the command specified by the *Command* parameter at the remote host specified by the *RemoteHost* parameter; if the *Command* parameter is not specified, the **rsh** command logs into the remote host specified by the *RemoteHost* parameter. The **rsh** command sends standard input from the local command line to the remote command and receives standard output and standard error from the remote command.

Note: Because any input to the remote command must be specified on the local command line, you cannot use the **rsh** command to execute an interactive command on a remote host. If you need to execute an interactive command on a remote host, use either the **rlogin** command or the **rsh** command without specifying the *Command* parameter. If you do not specify the *Command* parameter, the **rsh** command executes the **rlogin** command instead.

Access Files

If you do not specify the **-l** flag, the local user name is used at the remote host. If **-l** *User* is entered, the specified user name is used at the remote host.

Using Standard Authentication

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

- The local user ID is not the root user, and the name of the local host is listed as an equivalent host in the remote **/etc/hosts.equiv** file.
- If either the local user ID is the root user or the check of **/etc/hosts.equiv** is unsuccessful, the remote user's home directory must contain a **$HOME/.rhosts** file that lists the local host and user name.

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

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

For Kerberos 5 Authentication

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

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

Remote Command Execution

While the remote command is executing, pressing the Interrupt, Terminate, or Quit key sequences sends the corresponding signal to the remote process. However, pressing the Stop key sequence stops only the local process. Usually, when the remote command terminates, the local **rsh** process terminates.

To have shell metacharacters interpreted on the remote host, place the metacharacters inside " " (double quotes). Otherwise, the metacharacters are interpreted by the local shell.

When using the **rsh** command, you can create a link to a path (to which you have permission to write), using a host name specified by the *HostName* parameter as the link name. For example:

```
ln -s /usr/bin/rsh HostName
```

Having established this link, you can specify the *HostName* parameter and a command specified by the *Command* parameter from the command line, and the **rsh** command remotely executes the command on the remote host. The syntax is:

```
HostName Command
```

For example, if you are linked to remote host `opus` and want to perform the **date** command, enter:

```
opus date
```

Because you can not specify the **-l** *User* flag, the remote command is successful only if the local user has a user account on the remote host. Otherwise, the **rsh** command returns a `Login incorrect` error message. When you specify the *HostName* parameter without a command, the **rsh** command calls the **rlogin** command, which logs you into the remote host. Again, for successful login, the local user must have a user account on the remote host.

Flags

-a	Indicates the standard error of the remote command is the same as standard output. No provision is made for sending arbitrary signals to the remote process.
-l *User*	Specifies that the **rsh** command should log in to the remote host as the user specified by the *User* variable instead of the local user name. If this flag is not specified, the local and remote user names are the same.
-n	Specifies that the **rsh** command should not read from standard input.
-f	Causes the credentials to be forwarded. This flag will be ignored if Kerberos 5 is not the current authentication method. Authentication will fail if the current DCE credentials are not marked forwardable.
-F	Causes the credentials to be forwarded. In addition the credentials on the remote system will be marked forwardable (allowing them to be passed to another remote system). This flag will be ignored if Kerberos 5 is not the current authentication method. Authentication will fail if the current DCE credentials are not marked forwardable.
-k *realm*	Allows the user to specify the realm of the remote station if it is different from the local systems realm. For these purposes, a realm is synonymous with a DCE cell. This flag will be ignored if Kerberos 5 is not the current authentication method.

Security

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

- The local user ID is listed as a principal in the authentication database and had performed a **kinit** to obtain an authentication ticket.
- If a **$HOME/.klogin** file exists, it must be located in the local user's **$HOME** directory on the target system. The local user must be listed as well as any users or services allowed to **rsh** into this account. This file performs a similar function to a local **.rhosts** file. Each line in this file should contain a principal

in the form of *principal.instance@realm*. If the originating user is authenticated as one of the principals named in **.klogin**, access is granted to the account. The owner of the account is granted access if there is no **.klogin** file.

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

Exit Status

This command returns the following exit values:

0	Successful completion.
>0	An error occurred.

Examples

In the following examples, the local host, `host1`, is listed in the **/etc/hosts.equiv** file at the remote host, `host2`.

1. To check the amount of free disk space on a remote host, enter:

   ```
   rsh host2 df
   ```

   The amount of free disk space on `host2` is displayed on the local system.
2. To append a remote file to another file on the remote host, place the >> metacharacters in quotation marks, and enter:

   ```
   rsh host2 cat test1 ">>" test2
   ```

   The file `test1` is appended to `test2` on remote host `host2`.
3. To append a remote file at the remote host to a local file, omit the quotation marks, and enter:

   ```
   rsh host2 cat test2 >> test3
   ```

   The remote file `test2` on `host2` is appended to the local file `test3`.
4. To append a remote file to a local file and use a remote user's permissions at the remote host, enter:

   ```
   rsh host2  -l jane cat test4 >> test5
   ```

   The remote file `test4` is appended to the local file `test5` at the remote host, with user `jane`'s permissions.
5. This example shows how the root user can issue an **rcp** on a remote host when the authentication is Kerberos 4 on both the target and server. The root user must be in the authentication database and must have already issued **kinit** on the local host. The command is issued at the local host to copy the file, stuff, from node r05n07 to node r05n05 on an SP.

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

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

Files

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

/usr/lpp/ssp/rcmd/bin/remsh	Link to AIX Secure **/usr/bin/rsh** that calls the SP Kerberos 4 **rsh** routine if applicable.

Prerequisite Information

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

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

Related Information

The **ftp** command, **rcp** command, **rexec** command, **rlogin** command, **telnet**, **tn**, or **tn3270** command.

SP Commands: **k4init**, **k4list**, **k4destroy**, **lsauthpar**, **chauthpar**, **Kerberos**

Environment variable: **KRBTKFILE**

The **rshd** and **krshd** daemon.

The **kvalid_user** function.

The **hosts.equiv** file format, **.rhosts** file format.

Network Overview in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

Secure Rcmds in *AIX 5L Version 5.2 System User's Guide: Communications and Networks*.

---

rshd Daemon

Purpose

Provides the server function for remote command execution.

Syntax

Note: The **rshd** daemon is usually started by the **inetd** daemon. It can also be controlled from the command line, using SRC commands.

/usr/sbin/rshd [**-c**] [**-s**]

Description

The **/usr/sbin/rshd** daemon is the server for the **rcp** and **rsh** commands. The **rshd** daemon provides remote execution of shell commands. These commands are based on requests from privileged sockets on trusted hosts. The shell commands must have user authentication. The **rshd** daemon listens at the socket defined in the **/etc/services** file.

Changes to the **rshd** daemon can be made using the System Management Interface Tool (SMIT) or System Resource Controller (SRC), by editing the **/etc/inetd.conf** or /**etc**/**services** file. Entering `rshd` at the command line is not recommended. The **rshd** daemon is started by default when it is uncommented in the **/etc/inetd.conf** file.

The **inetd** daemon get its information from the /**etc**/**inetd.conf** file and the /**etc**/**services** file.

After changing the **/etc/inetd.conf** or /**etc**/**services** file, run the **refresh -s inetd** or **kill -1** *InetdPID* command to inform the **inetd** daemon of the changes to its configuration file.

Service Request Protocol

When the **rshd** daemon receives a service request, it initiates the following protocol:

1. The **rshd** daemon checks the source port number for the request. If the port number is not in the range 0 through 1023, the **rshd** daemon terminates the connection.
2. The **rshd** daemon reads characters from the socket up to a null byte. The string read is interpreted as an ASCII number (base 10). If this number is nonzero, the **rshd** daemon interprets it as the port number of a secondary stream to be used as standard error. A second connection is created to the specified port on the client host. The source port on the local host is also in the range 0 through 1023.
3. The **rshd** daemon uses the source address of the initial connection request to determine the name of the client host. If the name cannot be determined, the **rshd** daemon uses the dotted decimal representation of the client host's address.
4. The **rshd** daemon retrieves the following information from the initial socket:
   - A null-terminated string of at most 16 bytes interpreted as the user name of the user on the client host.
   - A null-terminated string of at most 16 bytes interpreted as the user name to be used on the local server host.
   - Another null-terminated string interpreted as a command line to be passed to a shell on the local server host.
5. The **rshd** daemon attempts to validate the user using the following steps:
   a. The **rshd** daemon looks up the local user name in the **/etc/passwd** and **/etc/security/passwd** file and checks to see if the password, and therefore the user, has expired. It also verifies that the encrypted password string is valid and tries to switch to the home directory (using the **chdir** subroutine). If either the lookup or the directory change fails, the **rshd** daemon terminates the connection.
   b. If the local user ID is a nonzero value, the **rshd** daemon searches the **/etc/hosts.equiv** file to see if the name of the client workstation is listed. If the client workstation is listed as an equivalent host, the **rshd** daemon validates the user.
   c. If the **$HOME/.rhosts** file exists, the **rshd** daemon tries to authenticate the user by checking the **.rhosts** file.
   d. If either the **$HOME/.rhosts** authentication fails or the client host is not an equivalent host, the **rshd** daemon terminates the connection.
6. After **rshd** validates the user, the **rshd** daemon returns a null byte on the initial connection and passes the command line to the user's local login shell. The shell then inherits the network connections established by the **rshd** daemon.

The **rshd** daemon should be controlled using the System Management Interface Tool (SMIT) or by changing the **/etc/inetd.conf** file. Typing `rshd` at the command line is not recommended.

Manipulating the rshd Daemon with the System Resource Controller

The **rshd** daemon is a subserver of the **inetd** daemon, which is a subsystem of the System Resource Controller (SRC). The **rshd** daemon is a member of the **tcpip** SRC subsystem group. This daemon is enabled by default in the **/etc/inetd.conf** file and can be manipulated by the following SRC commands:

startsrc	Starts a subsystem, group of subsystems, or a subserver.
stopsrc	Stops a subsystem, group of subsystems, or a subserver.
lssrc	Gets the status or a subsystem, group or subsystems, or a subserver.

Flags

-c Suppresses the reverse hostname lookup.
-s Turns on socket-level debugging.

Examples

Note: The arguments for the **rshd** daemon can be specified by using SMIT or by editing the **/etc/inetd.conf** file.

1. To start the **rshd** daemon, type the following:

   ```
   startsrc -t shell
   ```

   This command starts the **rshd** subserver.
2. To stop the **rshd** daemon, type the following:

   ```
   stopsrc -t shell
   ```

   This command allows all pending connections to start and existing connections to complete but prevents new connections from starting.
3. To force stop the **rshd** daemon and all **rshd** connections, type the following:

   ```
   stopsrc -t -f shell
   ```

   This command terminates all pending connections and existing connections immediately.
4. To display a short status report about the **rshd** daemon, type the following:

   ```
   lssrc -t shell
   ```

   This command returns the daemon's name, process ID, and state (active or inactive).

Related Information

The **kill** command, **lssrc** command, **refresh** command, **rsh** command, **startsrc** command, **stopsrc** command.

The **inetd** daemon.

The **$HOME/.rhosts** file format, **/etc/hosts.equiv** file format, **/etc/inetd.conf** file format, **/etc/services** file format.

The **rcmd** subroutine.

TCP/IP Daemons in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices*.

---

rstatd Daemon

Purpose

Returns performance statistics obtained from the kernel.

Syntax

/usr/sbin/rpc.rstatd

Description

The **rstatd** daemon is a server that returns performance statistics obtained from the kernel. The **rstatd** daemon is normally started by the **inetd** daemon.

Files

/etc/inetd.conf	TCP/IP configuration file that starts RPC daemons and other TCP/IP daemons.
/etc/services	Contains an entry for each server available through Internet.

Related Information

The **nfsstat** command.

The **inetd** daemon.

Network File System (NFS) Overview for System Management in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

List of NFS Commands.

---

rtl_enable Command

Purpose

Relinks shared objects to enable the runtime linker to use them. This command only applies to AIX 4.2 or later.

Syntax

rtl_enable [**-R** | **-o** *Name*] [**-l**] [**-s**] *File* [*ldFlag* ...] [**-F** *ObjsLibs* ...]

Description

The **rtl_enable** command relinks a module, or an archive containing modules, with the **-G** flag, to enable runtime linking. A module is an XCOFF file containing a loader section. A shared object is a module with the F_SHROBJ flag set in the XCOFF header.

In its simplest form, the **rtl_enable** command creates a new file with the name *File***.new**. If *File* is a module, *File***.new** will be the same kind of module. If *File* is an archive, *File***.new** will be an archive whose members have the same names as the members of *File*. The **rtl_enable** command relinks the modules in the new archive to enable run-time linking. The **rtl_enable** command archives other members unchanged into the output file.

The **rtl_enable** command uses the loader section in *File* (or its members) to create import and export files, to determine the **libpath** information, and to determine the entry point.

Flags

-F *ObjsLibs* ...	Adds *ObjsLibs* to the beginning of the generated **ld** command. The *ObjsLibs* parameter is either an object file or a library (specified with the **ld** command's **-l** (lowercase L) flag). If you are enabling an archive, adds the *ObjsLibs* to the **ld** command for all shared objects in the archive.

-l	(Lowercase L) Leaves the import and export files in the current directory instead of deleting them. Import files have the suffix **.imp** and export files, the suffix **.exp**. The **rtl_enable** command adds the suffixes to the input file name if *File* is a module. It adds the suffixes to the names of members that are modules if *File* is an archive.
-o *Name*	Specifies an alternate output file name instead of *File***.new**. Do not use this flag with the **-R** flag.
-R	Replaces the input file instead of creating a new file. It will not overwrite the input file if any errors occur. Do not use this flag with the **-o** flag.
-s	Generates a script of commands in the current directory that you can use to create a new output file or archive, but does not relink anything. It names the script *Base***.sh**, where *Base* is the basename of the input file with any suffix stripped off. It writes generated import and export files in the current directory as well. You can modify the script and the import and export files to customize the output objects.

Parameters

File	Specifies the input file.
ldFlag ...	Copies the specified **ld** command flags to the end of the generated **ld** command, overriding default options.

Note: Do not use the **-o** flag in the *ldFlag* parameter to name the output file. To specify an alternate output file name, use the **rtl_enable** command's **-o** *Name* flag.

Exit Status

This command returns the following exit values:

0	Successful completion.
>0	An error occurred.

Note: Depending on the error, some output files may have been created.

Security

Access Control: Any User

Auditing Events: N/A

Examples

To create a new version of **libc.a** with runtime linking enabled, enter:

1. Create a directory for runtime version by entering:
   ```
   mkdir /tmp/rtllibs
   ```
2. Make `/tmp/rtllibs` your current directory by entering:
   ```
   cd /tmp/rtllibs
   ```
3. To create the runtime version of libc.a with the same name, enter:
   ```
   rtl_enable -o libc.a /lib/libc.a
   ```

To use this version of libc.a when linking programs, use **-L /tmp/rtllibs** with the **ld** command.

Files

/usr/bin/rtl_enable	Contains the **rtl_enable** command. This is a symbolic link to **/usr/ccs/bin/rtl_enable**.

Related Information

Commands: ld.

Shared Objects and Runtime Linking in *AIX 5L Version 5.2 General Programming Concepts: Writing and Debugging Programs*.

---

runacct Command

Purpose

Runs daily accounting.

Syntax

/usr/sbin/acct/runacct [*mmdd* [*State*]]

Description

The **runacct** command is the main daily accounting shell procedure. Normally initiated by the **cron** daemon, the **runacct** command processes connect, fee, disk, queuing system (printer), and process accounting data files for the current day to produce the binary daily report, **/var/adm/acct/nite/dayacct**. The **runacct** command also prepares summary files for the **prdaily** procedure to prepare the ASCII daily report, **/var/adm/acct/sum/rprt***mmdd*, or for billing purposes.

The **acctmerg** command adds the **dayacct** report to the cumulative summary report for the accounting period, **/var/adm/acct/sum/tacct**. The **tacct** report is used by the **monacct** command to produce the monthly report, **/var/adm/acct/fiscal**.

This command has two parameters that must be entered from the keyboard should you need to restart the **runacct** procedure. The date parameter, *mmdd*, enables you to specify the day and month for which you want to rerun the accounting. The *State* parameter enables a user with administrative authority to restart the **runacct** procedure at any of its states. For more information on restarting **runacct** procedures and on recovering from failures, refer to ″Restarting **runacct** Procedures″.

The **runacct** command protects active accounting files and summary files in the event of run-time errors, and records its progress by writing descriptive messages into the **/var/adm/acct/nite/active** file. When the **runacct** procedure encounters an error, it sends mail to users root and adm, and exits.

The **runacct** procedure also creates two temporary files, **lock** and **lock1**, in the directory **/var/adm/acct/nite**, which it uses to prevent two simultaneous calls to the **runacct** procedure. It uses the **lastdate** file (in the same directory) to prevent more than one invocation per day.

The **runacct** command breaks its processing into separate, restartable states. As it completes each state, it writes the name of the next state in the **/var/adm/acct/nite/state** file. The **runacct** procedure processes the various states in the following order:

State	Actions
SETUP	Moves the active accounting files to working files and restarts the active files.
WTMPFIX	Verifies the integrity of the **wtmp** file, correcting date changes if necessary.
CONNECT1	Calls the **acctcon1** command to produce connect session records.

State	Actions
CONNECT2	Converts connect session records into total accounting records (**tacct.h** format).
PROCESS	Converts process accounting records into total accounting records (**tacct.h** format).
MERGE	Merges the connect and process total accounting records.
FEES	Converts the output of the **chargefee** command into total accounting records (**tacct.h** format) and merges them with the connect and process total accounting records.
DISK	Merges disk accounting records with connect, process, and fee total accounting records.
QUEUEACCT	Sorts the queue (printer) accounting records, converts them into total accounting records (**tacct.h** format), and merges them with other total accounting records.
MERGETACCT	Merges the daily total accounting records in the **daytacct** report file with the summary total accounting records in the **/var/adm/acct/sum/tacct** report file.
CMS	Produces command summaries in the file **/var/adm/acct/sum/cms**.
USEREXIT	If the **/var/adm/siteacct** shell file exists, calls it at this point to perform site-dependent processing.
CLEANUP	Deletes temporary files and exits.

Restarting runacct Procedures

To restart the **runacct** command after a failure, first check the **/var/adm/acct/nite/active** file for diagnostic messages, then fix any damaged data files, such as **pacct** or **wtmp**. Remove the **lock** files and **lastdate** file (all in the **/var/adm/acct/nite** directory), before restarting the **runacct** command. You must specify the *mmdd* parameter if you are restarting the **runacct** command. It specifies the month and day for which the **runacct** command is to rerun the accounting. The **runacct** procedure determines the entry point for processing by reading the **/var/adm/acct/nite/statefile** file. To override this default action, specify the desired *state* on the **runacct** command line.

It is not usually a good idea to restart the **runacct** command in the SETUP *state*. Instead, perform the setup actions manually and restart accounting with the WTMPFIX state, as follows:

```
/usr/lib/acct/runacct mmdd WTMPFIX
```

If the **runacct** command fails in the PROCESS state, remove the last **ptacct** file, because it will be incomplete.

Security

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

Examples

1. To start daily accounting procedures for system resources, add the following command line to a **crontab** file so the **runacct** command will be run automatically by the **cron** daemon:

   ```
   0 4 * * 1-6 /usr/sbin/acct/runacct 2> \
   /var/adm/acct/nite/accterr
   ```

   This example shows the instructions that the **cron** daemon will read and act upon. The **runacct** command will run at 4 a.m. (`04`) every Monday through Saturday (`1-6`) and write all standard error output (`2>`) to the **/var/adm/acct/nite/accterr** file. This command is only one of the accounting instructions normally given to the **cron** daemon. See ″Setting Up an Accounting System″ in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices* for more information on typical **cron** accounting entries.
2. To start daily accounting procedures for system resources from the command line (start the **runacct** command), enter the following:

   ```
   nohup /usr/sbin/acct/runacct 2> \
   /var/adm/acct/nite/accterr &
   ```

   Although it is preferable to have the **cron** daemon start the **runacct** procedure automatically (see example 1), you can give the command from the keyboard. The **runacct** command will run in the

background (&), ignoring all INTERRUPT and QUIT signals (the **nohup** command), and write all standard error output (2>) to the **/var/adm/acct/nite/accterr** file.

3. To restart the system accounting procedures for a specific date, enter a command similar to the following:

   ```
   nohup /usr/sbin/acct/runacct 0601 2>> \
   /var/adm/acct/nite/accterr &
   ```

   This example restarts **runacct** for the day of June 1 (0601). The **runacct** command reads the file **/var/adm/acct/nite/statefile** to find out the state with which to begin. The **runacct** command will run in the background (&), ignoring all INTERRUPT and QUIT signals (**nohup**). Standard error output (2) is added to the end (>>) of the **/var/adm/acct/nite/accterr** file.

4. To restart the system accounting procedures for a particular date at a specific state, enter a command similar to the following:

   ```
   nohup /usr/sbin/acct/runacct 0601 MERGE 2>> \
    /var/adm/acct/nite/accterr &
   ```

   This example restarts the **runacct** command for the day of June 1 (0601), starting with the MERGE state. The **runacct** command will run in the background (&), ignoring all INTERRUPT and QUIT signals (the **nohup** command). Standard error output (2) is added to the end (>>) of the **/var/adm/acct/nite/accterr** file.

Files

/var/adm/wtmp	Log in/log off history file.
/var/adm/pacct*	Process accounting file.
/var/adm/acct/nite/daytacct	Disk usage accounting file.
/var/adm/qacct	Active queue accounting file.
/var/adm/fee	Record of fees charged to users.
/var/adm/acct/sum/*	Command and total accounting summary files.
/var/adm/acct/nite/ptacct*.*mmdd*	Concatenated version of **pacct** files.
/var/adm/acct/nite/active	The **runacct** message file.
/var/adm/acct/nite/lock*	Prevents simultaneous invocation of **runacct**.
/var/adm/acct/nite/lastdate	Contains last date **runacct** was run.
/var/adm/acct/nite/statefile	Contains current state to process.

Related Information

The **acctcms** command, **acctcom** command, **acctcon1** or **acctcon2** command, **acctmerg** command, **acctprc1**, **acctprc2**, or **accton** command, **crontab** command, **fwtmp** command.

The **cron** daemon.

The **acct** subroutine.

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

For more information about the Accounting System, the preparation of daily and monthly reports, and the accounting files, see the Accounting Overview in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices*.

Setting Up an Accounting System in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices* describes the steps you must take to establish an accounting system.

See the Accounting Commands in *AIX 5L Version 5.2 System Management Guide: Operating System and Devices* for a list of accounting commands that can be run automatically or entered from the keyboard.

runcat Command

Purpose
Pipes output data from the **mkcatdefs** command to the **gencat** command.

Syntax
runcat *CatalogName SourceFile* [*CatalogFile*]

Description
The **runcat** command invokes the **mkcatdefs** command and pipes the message catalog source data (the output from **mkcatdefs**) to the **gencat** program.

The file specified by the *SourceFile* parameter contains the message text with your symbolic identifiers. The **mkcatdefs** program uses the *CatalogName* parameter to generate the name of the symbolic definition file by adding **_msg.h** to the end of the *CatalogName* value, and to generate the symbolic name for the catalog file by adding **MF_** to the beginning of the *CatalogName* value. The definition file must be included in your application program. The symbolic name for the catalog file can be used in the library functions (such as the **catopen** subroutine).

The *CatalogFile* parameter is the name of the catalog file created by the **gencat** command. If you do not specify this parameter, the **gencat** command names the catalog file by adding **.cat** to the end of the *CatalogName* value. This file name can also be used in the **catopen** library function.

Example
To generate a catalog named `test.cat` from the message source file `test.msg`, enter:

```
runcat test test.msg
```

File

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

Related Information
The **dspcat** command, **dspmsg** command, **gencat** command, **mkcatdefs** command.

The **catclose** subroutine, **catgets** subroutine, **catopen** subroutine.

Message Facility in *AIX 5L Version 5.2 National Language Support Guide and Reference*.

rup Command

Purpose
Shows the status of a remote host on the local network.

Syntax
/usr/bin/rup [**-h** | **-l** | **-t**] [*Host ...*]

Description
The **rup** command displays the status of a remote host by broadcasting on the local network and then displaying the responses it receives. Specify a flag if you want to sort the output. If you do not specify a flag, the **rup** command displays responses in the order they are received. If you specify multiple hosts on

the command line, the **rup** command ignores any flags and displays output in the order you specified the hosts. You must use the **sort** command to sort the output.

In addition, when you provide a value for the *Host* parameter, the **rup** command queries the hosts you specify, rather than broadcasting to all hosts. A remote host responds only if it is running the **rstatd** daemon, which is normally started from the **inetd** daemon.

Notes:

1. Broadcasting does not work through gateways. Therefore, if you do not specify a host, only hosts on your network can respond to the **rup** command.
2. Load-average statistics are not kept by the kernel. The load averages are always reported as 0 (zero) by this command.

Flags

-h Sorts the display alphabetically by host name.
-l Sorts the display by load average.
-t Sorts the display by length of runtime on the network.

Examples

1. To find out the status of all hosts on the network and to sort the list alphabetically by host name, enter:

   ```
   /usr/bin/rup -h
   ```

2. To display a list of all hosts on the network according to each machine's load average, enter:

   ```
   /usr/bin/rup -l
   ```

3. To display the status of a host, enter:

   ```
   /usr/bin/rup brutus
   ```

   In this example, the **rup** command displays the status of the host named `brutus`.

4. To display the status of all hosts on the network sorted by each machine's length of runtime, enter:

   ```
   /usr/bin/rup -t
   ```

Files

/etc/inetd.conf Defines how the **inetd** daemon handles Internet service requests.

Related Information

The **sort** command, the List of NFS Commands.

The **inetd** daemon, **rstatd** daemon.

Network File System (NFS) Overview for System Management in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

---

ruptime Command

Purpose

Shows the status of each host on a network.

Syntax

ruptime [**-a**] [**-r**] [**-l** | **-t** | **-u**]

Description

The **/usr/bin/ruptime** command displays the status of each host that is on a local network and is running the **rwhod** daemon. The status lines are sorted by host name unless the **-l**, **-t**, or **-u** flag is indicated. The status information is provided in packets broadcast once every 3 minutes by each network host running the **rwhod** daemon. Any activity (such as power to a host being turned on or off) that takes place between broadcasts is not reflected until the next broadcast. Hosts for which no status information is received for 11 minutes are reported as down.

Output is in the following format: hostname, status, time, number of users, and load average. Load average represents the load averages over 1-, 5-, and 15-minute intervals prior to a server's transmission. The load averages are multiplied by 10 to represent the value in decimal format.

Flags

-a Includes all users. Without this flag, users whose sessions are idle an hour or more are not included.
-l Sorts the list by the load average.
-r Reverses the sort order. The **-r** flag should be used with the **-l**, **-t** or **-u** flag.
-t Sorts the list by the uptime.
-u Sorts the list by the number of users.

Examples

1. To get a status report on the hosts on the local network, enter:

   ```
   ruptime
   ```

   Information similar to the following is displayed:

   ```
   host1      up      5:15,   4 users,   load 0.09, 0.04, 0.04
   host2      up      7:45,   3 users,   load 0.08, 0.07, 0.04
   host7      up      7:43,   1 user,    load 0.06, 0.12, 0.11
   ```

2. To get a status report sorted by load average, enter:

   ```
   ruptime  -l
   ```

   Information similar to the following is displayed:

   ```
   host2      up      7:45,   3 users,   load 0.08, 0.07, 0.04
   host1      up      5:18,   4 users,   load 0.07, 0.07, 0.04
   host7      up      7:43,   1 user,    load 0.06, 0.12, 0.11
   ```

Files

/var/spool/rwho/whod.* Indicates data files received from remote **rwhod** daemons.

Related Information

The **rwho** command.

The **rwhod** daemon.

Network Overview in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

ruser Command

Purpose

Directly manipulates entries in three separate system databases that control foreign host access to programs.

Syntax

To Add or Delete a Database File Name Entry

ruser { **-a** | **-d** } { **-f** *"UserName ..."* | **-p** *"HostName ..."* | **-r** *"HostName ..."* }

To Delete or Display All Name Entries in a Database File

ruser { **-X** | **-s** } { **-F** | **-P** | **-R** }

Description

The **ruser** low-level command adds or deletes entries in three separate system databases. Which database you are manipulating is determined by using the **-p**, **-r**, or **-f** flags. In addition, the **ruser** command can show one or all entries in one of the databases. Each database is a list of names. The three databases are as follows:

- **/etc/ftpusers** file
- **/etc/hosts.equiv** file
- **/etc/hosts.lpd** file

Note: The **-p** and **-r** options can be used together to add a name to databases at the same time, but the **-f** option cannot be used with either.

You can use the Users application in Web-based System Manager to change user characteristics.

You could also use the System Management Interface Tool (SMIT) **smit users** fast path to run this command or type:

```
smit rprint
```

Flags

-a	Adds a name to the database. The **-a** flag must be used with either the **-p**, **-r**, or **-f** flag.
-d	Deletes a name from the database. Must be used with either the **-p**, **-r**, or **-f** flag.
-F	Deletes or shows all entries in the **/etc/ftpusers** file. Use this flag with the **-X** flag to delete all entries. Use this flag with the **-s** flag to show all entries.
-f *"UserName ..."*	Adds or deletes the user name specified by the *UserName* variable to the **/etc/ftpusers** database that contains a list of local user names that cannot be used by remote FTP clients. The **-f** flag must be used with either the **-a** or **-d** flag.
-P	Deletes or shows all entries in the **/etc/hosts.lpd** file. Use this flag with the **-X** flag to delete all entries. Use this flag with the **-s** flag to show all entries.
-p *"HostName ..."*	Adds or deletes the host name, specified by the *HostName* variable, in the database that specifies which foreign host may print on your machine. The **-p** flag must be used with either the **-a** or **-d** flag.
-R	Deletes or shows all entries in the **/etc/hosts.equiv** file. Use this flag with the **-X** flag to delete all entries. Use this flag with the **-s** flag to show all entries.
-r *"HostName ..."*	Adds or deletes the host name, specified by the *HostName* variable, in the **/etc/hosts.equiv** database that specifies which foreign host may perform the remote commands (**rlogin**, **rcp**, **rsh**, or **print**) on your machine. The **-r** flag must be used with either the **-a** or **-d** flag.

-s	Shows all entries in the database. Use this flag with either the **-P**, **-R**, or **-F** flag.
-X	Deletes all names from the database. Use this flag with either the **-P**, **-R**, or **-F** flag.

Examples

1. To add an entry in the **/etc/hosts.lpd** database, which specifies which foreign host may print on the local machine, type the command in the following format:

   ```
   ruser  -a  -p "host1"
   ```

   In this example, the foreign host is `host1`.
2. To delete an entry in the database that controls printing only (**/etc/hosts.lpd**), and also delete the same name from the database that controls remote access for the **rlogin**, **rcp**, and **rsh** commands (**/etc/hosts.equiv**), type:

   ```
   ruser  -d  -r "host2"  -p "host1"
   ```

   In this example, the host from which the database entry is deleted is `host1`.

Related Information

The **lpd** daemon, **rshd** daemon, **rlogind** daemon.

The **ftpusers** file format, **hosts.equiv** file format, **hosts.lpd** file format.

For information on installing the Web-based System Manager, see Chapter 2: Installation and System Requirements in *AIX 5L Version 5.2 Web-based System Manager Administration Guide*.

Understanding the TCP/IP Reference in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

---

rusers Command

Purpose

Reports a list of users logged on to remote machines.

Syntax

/usr/bin/rusers [**-a**] [**-l**] [**-u** | **-h** | **-i**] [*Host ...*]

Description

The **rusers** command produces a list of users who are logged on to remote machines. The **rusers** command does this by broadcasting to each machine on the local network and printing the responses it receives. Normally, the system prints the responses in the order they are received. To change this order, specify one of the flags. In addition, when you provide a *Host* parameter, the **rusers** command queries the host or hosts you specify, rather than broadcasting to all hosts.

By default, each entry contains a list of users for each machine. Each of these entries includes the names of all users logged in that machine. In addition, when the user does not type into the system for a minute or more, the **rusers** command reports the user's idle time.

A remote host responds only if it is running the **rusersd** daemon, which is normally started up from the **inetd** daemon.

Note: Broadcasting does not work through gateways. Therefore, if you do not specify a host, only hosts on your network can respond to the **rusers** command.

Flags

-a Gives a report for a machine even if no users are logged in.
-h Sorts alphabetically by host name.
-i Sorts by idle time.
-l Gives a longer listing similar to the **who** command.
-u Sorts by number of users.

Examples

1. To produce a list of the users on your network that are logged in remote machines, enter:

   ```
   rusers
   ```

2. To produce a list of users sorted alphabetically by host name, enter:

   ```
   rusers -h
   ```

3. To produce a list of users on a host, enter:

   ```
   rusers -h pluto
   ```

   In this example, the **rusers** command produces a list of users on the host named `pluto`.

4. To produce a list of users logged in remote machines and sorted according to each machine's length of idle time, enter:

   ```
   rusers -i
   ```

5. To produce a list of users logged in remote machines and sorted by the number of users logged in, enter:

   ```
   rusers -u
   ```

Files

/etc/inetd.conf
TCP/IP configuration file that starts RPC daemons and other TCP/IP daemons.

Related Information

The **rwho** command, **who** command.

The **inetd** daemon, **rusersd** daemon.

Network File System (NFS) Overview for System Management in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

List of NFS Commands.

---

rusersd Daemon

Purpose

Responds to queries from the **rusers** command.

Syntax

/usr/lib/netsvc/rusers/rpc.rusersd

Description

The **rusersd** daemon is a server that responds to queries from the **rusers** command by returning a list of users currently on the network. This daemon is normally started by the **inetd** daemon.

Files

/etc/inetd.conf	TCP/IP configuration file that starts RPC daemons and other TCP/IP daemons.
/etc/utmp	Contains information on users logged in to the system.

Related Information

The **rusers** command.

The **inetd** daemon.

Network File System (NFS) Overview for System Management in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

List of NFS Commands.

---

rwall Command

Purpose

Sends messages to all users on the network.

Syntax

To Send a Message to Specified Hosts

/usr/sbin/rwall *HostName* ...

To Send a Message to Specified Networks

/usr/sbin/rwall -n *NetworkGroup* ...

To Send a Message to Specified Hosts on a Network

/usr/sbin/rwall -h *HostName* ... **-n** *NetworkGroup*

Description

The **rwall** command sends messages to all users on the network. To do this, the **rwall** command reads a message from standard input until it reaches an end-of-file character. The **rwall** command takes this message, which begins with the line `Broadcast Message...`, and broadcasts it to all users logged in to the specified host machines. Users receive messages only if they are running the **rwalld** daemon, which is started by the **inetd** daemon.

> **Note:** The time out is fairly short. This enables the **rwall** command to send messages to a large group of machines (some of which may be down) in a reasonable amount of time. Thus the message may not get through to a heavily loaded machine.

Flags

-h Sends the message to machines specified by the *HostName* parameter.

-n Sends the message to specific network groups only. Network groups are defined in the **netgroup** file.

Examples

1. To send a message to a host named `neptune`, enter:
   ```
   /usr/sbin/rwall neptune
   ```
   Type in your message. When you are done, enter:
   ```
   Ctrl D
   ```
2. To send a message to a host named `neptune` and every host in the `cosmos` netgroup, enter:
   ```
   rwall  -n cosmos  -h neptune
   ```
   Type in your message. When you are done, enter:
   ```
   Ctrl D
   ```

Files

/etc/inetd.conf	TCP/IP configuration file that starts RPC daemons and other TCP/IP daemons.
/etc/netgroup	Contains information about each user group on the network.

Related Information

The **wall** command.

The **inetd** daemon, **rwalld** daemon.

Network File System (NFS) Overview for System Management in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

List of NFS Commands.

---

rwalld Daemon

Purpose

Handles requests from the **rwall** command.

Syntax

/usr/lib/netsvc/rwall/rpc.rwalld

Description

The **rwalld** daemon handles requests from the **rwall** command. The **inetd** daemon invokes the **rwalld** daemon.

Files

/etc/inetd.conf	Specifies the TCP/IP configuration.

Related Information

The **rwall** command.

The **inetd** daemon.

Network File System (NFS) Overview for System Management in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

List of NFS Commands.

---

rwho Command

Purpose

Shows which users are logged in to hosts on the local network.

Syntax

rwho [**-a**]

Description

The **/usr/bin/rwho** command displays the user name, host name, and start date and time of each session for everyone on the local network who is currently logged in to a host running the **rwhod** daemon. If a workstation is inactive for at least 3 minutes, the **rwho** command reports the idle time as a number of minutes in the last column. After an hour of inactivity, a user is not included unless the **-a** flag is specified.

> **Note:** Since this command displays a lot of output, use this command with caution if the local network has a large number of users.

Status information is broadcast once every 3 minutes by each network host running the **rwhod** daemon. Any activity (such as a user logging on or off) that takes place between broadcasts is not reflected until the next broadcast.

Flags

-a Includes all users. Without this flag, users whose sessions are idle an hour or more are not included in the report.

Example

To get a report of all users currently logged in to hosts on the local network, enter:

```
rwho
```

Information similar to the following is displayed:

```
bob     host2:pts5       Nov 17 06:30 :20
bob     host7:console    Nov 17 06:25 :25
fran    host1:pts0       Nov 17 11:20 :51
fran    host1:pts8       Nov 16 15:33 :42
fran    host4:console    Nov 17 16:32
server  host2:console    Nov 17 06:58 :20
alice   host2:pts6       Nov 17 09:22
```

Files

/var/spool/rwho/whod.* Indicates data files received from remote **rwhod** daemons.

Related Information

The **ruptime** command, **rwho** command, **who** command.

The **gethostname** subroutine.

The **services** file format.

Network Overview in *AIX 5L Version 5.2 System Management Guide: Communications and Networks*.

---

rwhod Daemon

Purpose

Provides the server function for the **rwho** and **ruptime** commands.

Syntax

Note: Use SRC commands to control the **rwhod** daemon from the command line. Use the **rc.tcpip** file to start the daemon with each system startup.

/usr/sbin/rwhod

Description

The **/usr/sbin/rwhod** daemon maintains the database used by the **rwho** and **ruptime** commands. Once started, the **rwhod** daemon operates as both producer and consumer of status information.

As a producer of status information, the **rwhod** daemon queries the state of the local host approximately every 3 minutes. It then constructs status messages and broadcasts them to the local network.

As a consumer of status information, the **rwhod** daemon listens for status messages from **rwhod** servers on remote hosts. When the **rwhod** daemon receives a status message, it validates the received status message. It then records the message in the **/var/spool/rwho** directory. (The **rwho** and **ruptime** commands use the files in the **/var/spool/rwho** directory to generate their status listings.)

The **rwhod** daemon broadcasts and receives status messages using the **rwho** socket as specified in the **/etc/services** file.

When creating these messages, the **rwhod** daemon calculates the entries for the average CPU load for the previous 1-, 5-, and 15-minute intervals. Before broadcasting these messages, the **rwhod** daemon converts them to the byte order that the network can use.

When the **rwhod** daemon receives messages on the **rwho** socket, it discards any that do not originate from an **rwho** socket. Additionally, it discards any messages that contain unprintable ASCII characters. When the **rwhod** daemon receives a valid message, it places the message in a **whod.***HostName* file in the **/var/spool/rwho** directory, overwriting any file with the same name.

The **rwhod** daemon should be controlled using the System Resource Controller (SRC). Entering `rwhod` at the command line is not recommended.

Manipulating the rwhod Daemon with the System Resource Controller

The **rwhod** daemon is a subsystem controlled by the System Resource Controller (SRC). The **rwhod** daemon is a member of the **tcpip** system group. This daemon is disabled by default and can be manipulated by the following SRC commands:

stopsrc	Stops a subsystem, group of subsystems, or a subserver.
traceson	Enables tracing of a subsystem, group of subsystems, or a subserver.
tracesoff	Disables tracing of a subsystem, group of subsystems, or a subserver.
lssrc	Gets the status of a subsystem, group of subsystems, or a subserver.

Examples

1. To start the **rwhod** daemon, enter the following:

   ```
   startsrc -s rwhod
   ```

   This command starts the daemon. You can use this command in the **rc.tcpip** file or on the command line. The **-s** flag specifies that the subsystem that follows is to be started.
2. To stop the **rwhod** daemon normally, enter the following:

   ```
   stopsrc -s rwhod
   ```

   This command stops the daemon. The **-s** flag specifies that the subsystem that follows is to be stopped.
3. To get a short status report from the **rwhod** daemon, enter the following:

   ```
   lssrc -s rwhod
   ```

   This command returns the name of the daemon, the process ID of the daemon, and the state of the daemon (active or inactive).
4. To enable tracing for **rwhod** daemon, enter the following:

   ```
   traceson -s rwhod
   ```

   This command enables socket level debugging. Use the **trpt** command to look at the output of this example command.

Files

/etc/utmp	Contains status information on users that are logged in to the local host.
/var/spool/rwho/*	Contains files used by the **rwho** and **ruptime** commands to generate their status list.
/var/spool/rwho/whod.*HostName*	Contains the latest status information for the host specified by the *HostName* parameter.

Related Information

The **ruptime** command, **rwho** command, **who** command.

The **gethostname** subroutine.

The **services** file format.

TCP/IP Daemons in *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices*.

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, 2002

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

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

AIX

AIX 5L

IBM

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be the trademarks or service marks of others.

Index

Special characters

/etc/filesystems file
  removing entries
    using rmfs command 578
/etc/inittab file
  removing records
    using rmitab command 581
/etc/vfs file
  removing entries
    using rmvfs command 619

A

accounting system
  creating basic files
    using nulladm command 219
  deleting files
    using remove command 515
  displaying session records 351
  formatting ASCII report of previous day
    using prdaily command 352
  formatting total accounting records
    using prtacct command 386
  printer records
    preparing 237
  running
    using runacct command 653
acct/* commands
  nulladm 219
  prctmp 351
  prdaily 352
  prtacct 386
  remove 515
alias database
  building 40
asynchronous connection
  PPP subsystem
    pppdial command 344
asynchronous device stream
  attaching to PPP subsystem
    using pppattachd daemon 338
authenticate a user 243

B

backing up files
  onto a remote machines' device
    using rdump command 495
bibliographic database
  printing
    using roffbib command 624
boot image 527
Bourne shell
  invoking restricted version of
    using Rsh command 643

C

C programming language
  compiling patterns into declarations 512
characters
  reversing in each line of a file
    using rev command 540
command
  running without hangups 191
commands
  namerslv 9
  netstat 30
  newform 40
  nfso 49
  nim 69
  nimconfig 87
  niminit 93
  niscat 103
  nisdefaults 113
  nistbladm 139
  nlssrc 149
  no 154
  ntpq 212
  oslevel 234
  passwd 245
  pic 275
  pkgtrans 328
  pr 348
  preprpnode 352
  printf 357
  prs 383
  ps 391
  psroff 414
  ptfpkg 420
  qdaemon 442
  quot 463
  quota 465
  quotacheck 466
  quotaoff 467
  quotaon 467
  ras_logger 469
  rc 471
  rc.powerfail 473
  rcp 474
  rdist 481
  rdump 495
  read 497
  refrsrc 510
  remsh 645
  rev 540
  rmaudrec 557
  rmcctrl 563
  rmcomg 566
  rmcondition 568
  rmcondresp 570
  rmdel 573
  rmdev 574
  rmdir 575

© Copyright IBM Corp. 1997, 2002

commands *(continued)*
rmm 585
rmnamsv 586
rmnfsexp 588
rmramdisk 597
rmresponse 598
rmrpdomain 601
rmrpnode 604
rmrsrc 606
rmsensor 608
rmss 611
rmvirprt 620
rsh 645
Rsh 643
ruser 659
setting a different priority for
using nice command 67
configuration database
manipulating domain name server entries for local resolver routines in
using namerslv command 9
configuring
NFS network variables
using nfso command 49
NIM client package
using niminit command 93
copies of files, identical
on multiple hosts, distributes
using rdist command 481
creating a message source file 656
creating a PTF image
for PTFs in ptf_pkg.$BLDCYCLE
using ptfpkg command 420

D

daemons
ndpd-router 14
nrglbd 195
pcnfsd 635
pppattachd 338
routed 628
rshd 648
databases, system
controlling foreign host access, manipulating
using ruser command 659
default values
active in the namespace, displaying seven
using nisdefaults command 113
delayed login ports 269
delta files
removing
using rmdel command 573
device
removing from the system
using rmdev command 574
device configuration commands
restbase 527
Diablo 630 print file
converting to PostScript
using ps630 command 408
diff listings 250
directories
deleting 551
path name of
displaying 428
removing 551
using rmdir command 575
directory, unexports a
from NFS clients
using rmnfsexp command 588
disk usage
displaying quotas and
using quota command 465
displaying
network status
using netstat command 30
displaying files
SCCS
using prs command 383
displays
files
formatting to screen 271
distributing identical copies of files
on multiple hosts
using rdist command 481
document
finding literature references in
using refer command 505
inserting literature references in
using refer command 505
domain name server
entries for local resolver routines in configuration DB, manipulating
using namerslv command 9
draw pictures
preprocessing troff command input to
using pic command 275
drivers
formatting a printer
using pioformat command 298
dynamic host configuration protocol
update the DNS server
nsupdate command 204

E

echo request
sending to a network host
using ping command 284
enq command
scheduling jobs enqueued by the
using qdaemon command 442
environment
printing variable values 356

F

fastboot command 499
fields, standard input
compared to shell variables
using read command 497

file
checking the pathname
using pathchk command 255
compressing folder contents into a
using packf command 240
copying created by the backup command
using restore command 528
merging the lines in one
using paste command 248
reversing characters in each line of a
using rev command 540
file system
constructing a prototype file
using proto command 380
copying backup from remote machine to local machine
using rrestore command 640
ownership, summarizes
using quot command 463
quota consistency, checking
using quotacheck command 466
quotas, turning on and off
using quotaon or quotaoff command 467
files
backing up onto a remote machines' device
using rdump command 495
compressing
using pack command 238
deleting 551
displaying contents 271
displaying in specified format 221
displaying SCCS
using prs command 383
formatting to the display 271
generating path names from i-node numbers
using ncheck command 12
merging the lines of several
using paste command 248
numbering lines 147
ptf_pkg.$BLDCYCLE file
using ptfpkg command 420
removing 551
transferring between local and a remote host
using rcp command 474
transferring between two remote hosts
using rcp command 474
unpacking
using pcat command 268
writing to standard output
using pr command 348
folder
compressing into a file
using packf command 240
folders
deleting
using rmf command 577
removing messages within
using rmf command 577
fonts
copying
using piofontin command 297
foreign host access
manipulating entries in database, controlling
using ruser command 659
format
changing text
using newform command 40
formatted output
writing to standard output
using printf command 357

G

games
number-writing game 220
quiz 462
group ID
changing the primary
using newgrp command 43
groups
removing
using rmgroup command 580

H

HCON
System/370 Host Interface Adapter
diagnosing activity 244
hlppiomsg 309
hlppppdial 344
hlprestvg 538
host
showing the status on a
using ruptime command 657
hosts
connecting local with remote
using rlogin command 547
executing a command on a remote
using rsh command 645
logging into a remote
using rsh command 645
hosts, multiple
distributing identical copies of files on
using rdist command 481

I

identical copies of files
on multiple hosts, distributes
using rdist command 481
idformat 407, 408, 413, 414
idjprint 452
idposix 515
idsec 523
IMAP commands
pop3d 333
incoming messages
notifying user 481
sending copy to additional recipients 478
index
creating subject-page index
using ndx command 19

index *(continued)*
generating permuted
using ptx command 424
initialization, startup
using rc command 471
initializing
NIM master package
using nimconfig command 87
installable packages,
producing
using pkgmk command 321
Internet
querying domain name servers 198
IPv6 neighbor discovery protocol 13

L

libraries
converting archives 469
line printer
formatting text for
using nroff command 195
links
removing 551
local resolver routines
in configuration DB, manipulating domain name server entries for
using namerslv command 9
logical volume
copying one volume to a new volume 503
removing from a volume group
using rmlv command 583
removing mirrors from
using rmlvcopy command 584
login ports
delayed
enabling 269
listing 269
disabling
using pdisable command 269
using phold command 274
enabling
using pdelay command 269
using penable command 270
using pshare command 412
using pstart command 417
listing
using pdelay command 269
using penable command 270
using phold command 274
using pstart command 417
shared
enabling 412
listing 412

M

Mail commands
newaliases 40
rmail 557
mail, incorporation into folder, MH 479
maintenance level, system
reports latest installed
using oslevel command 234
mathematical text
formatting
using neqn command 20
memory management
displaying system page size 242
memory, various sizes
performance testing of applications, simulating a system with
using rmss command 611
message
showing the previous
using prev command 355
message facility commands
runcat 656
message routing 337
message sequences
creating
using pick command 281
modifying
using pick command 281
message source file
creating 656
messages
filing in other folders
using refile command 507
removing from active status
using rmm command 585
replying to
using repl command 518
saving in packed files 479
selecting by content
using pick command 281
showing the next
using next command 47
writing to standard output 45
MH
post command 337
prompter command 379
rcvdist command 478
rcvpack command 479
rcvstore command 479
rcvtty 481
MultiPath I/O
rmpath command 591
multiple hosts
distributing identical copies of files on
using rdist command 481

N

name command 89
name services
unconfiguring TCP/IP
using rmnamsv command 586
named daemon 1
description of 1
named9 8
namerslv command 9

namespace
displaying the seven default values active in
using nisdefaults command 113
National Television Standards Committee video output 218
ncheck command 12
NCS daemons
nrglbd 195
NDP and RIPng daemon
for a router
using ndpd-router daemon 14
ndp Command 13
ndp daemon 13
ndpd-router daemon 14
ndx command 19
neighbor discovery protocol 13
neqn command 20
nesgrep information
searching 116
netpmon Command 21
netstat command 30
interface display 33
routing table display 33
network CPU usage 21
Network File System
configuring network variables
using nfso command 49
Network Install Management
operations
using nim command 69
Network Install Manager 84, 87
configuring client package
using niminit command 93
initializes master package
using nimconfig command 87
network parameters
tuning
using no command 154
network routing tables
managing
using routed daemon 628
network status
displaying
using netstat command 30
Network Time Protocol
starting
using ntpq command 212
Network Time Protocol command
ntpdate 211
ntptrace 217
network variables
configuring NFS
using nfso command 49
newaliases command
Mail 40
newform command 40
newgrp command 43
newkey command
NIS 44
next command 47
NFS
configuring network variables
using nfso command 49
NFS clients
unexports a directory from
using rmnfsexp command 588
NFS commands
nfsstat 64
on 233
rmnfs 587
rmnfsexp 588
rmnfsmnt 589
rpcgen 637
rpcinfo 638
rup 656
rusers 660
rwall 662
NFS daemons
nfsd 48
pcnfsd 635
portmap 334
rexd 542
rstatd 650
rusersd 661
rwalld 663
nfsd daemon 48
nfso command 49
nfsstat command 64
nice command 67
NIM
configuring client package
using niminit command 93
initializes master package
using nimconfig command 87
nim command 69
NIM commands
nim 69
nimclient 84
nimconfig 87
niminit 93
NIM objects
performing operations
using nim command 69
nimadm command 79
nimclient command 84
nimconfig command 87
niminit command 93
NIS
commands
nistest 143
NIS commands
newkey 44
rm_niscachemgr 553
rm_nisd 554
rm_nispasswdd 556
rmkeyserv 581
rmyp 621
NIS+
table, displaying
using niscat command 103

NIS+ table
displaying
using niscat command 103
NIS+ tables
administering
using nistbladm command 139
niscat command 103
nisdefaults command 113
nisgrep command 116
nistbladm command 139
nistest command
description of 143
nistoldif 144
nlssrc command 149
no command 154
nohup command 191
notify object class (SRC)
removing a notify method definition 590
notifyevent script 193
nrglbd
description of 195
nroff command 195
formatting mathematical text for
using neqn command 20
nslookup command 198
nsupdate command 204
nsupdate9 208
NTP
starting
using ntpq command 212
ntpdate command 211
ntpq command 212
ntptrace command 217
ntsc Command 218
nulladm command 219

O

object files
displaying profile data 376
displaying symbol tables 151
obtaining
network status
using netstat command 30
ODM
compacts database 296
ODM (Object Data Manager)
object classes
adding objects to 225
changing objects in 227
compiling 227
deleting 231
displaying description 232
removing objects 230
retrieving objects matching criteria 231
objects
adding to object classes 225
changing in object classes 227
removing from object classes 230
retrieving matching criteria from object classes 231
on command 233
oslevel command 234
output, standard
writing to
using pr command 348
ownership
summarizes file system
using quot command 463

P

pac command 237
pack command 238
package format
translating
using pkgtrans command 328
packages,
producing installable
using pkgmk command 321
packf command 240
pagdel 242
paging space
removing 594
paginit 243
paglist 244
panel 20 Command 244
passwd command 245
password
changing
using passwd command 245
verifying information in the user database about a
using pwdck command 431
passwords
administering user
using pwdadm command 429
paste command 248
patch command 250
path name
displaying 428
pathchk command 255
pax command
extracts, writes, and lists archives and copies files 256
PC-NFS clients
handling service requests from
using pcnfsd daemon 635
pcat command 268
pcnfsd daemon 635
pdelay command 269
pdisable command 269
penable command 270
performance testing
simulating a system for
using rmss command 611
permuted index
generating
using ptx command 424
personal printer data stream
formatting text for printers with
using proff command 378
phold command 274

physical volume 522
pic command 275
pick command 281
pictures, draw
  preprocessing troff command input to
    using pic command 275
ping command
  description of 284
  example of 286
piobe command 290
pioburst command 292
piocnvt command 289, 294
piodigest command 295
piofontin command 297
pioformat command 298
piofquote command 301
pioout command 310
piopredef command 312
pkgmk command 321
pkgtrans command 328
plot file
  converting to PostScript
    using psplot command 413
plotgbe Command 329
plotlbe Command 331
plotter queue
  removing from the system
    using rmquedev command 596
plotting HP-GL files 329, 331
pop3d daemon 333
portmap daemon 334
portmir command 335
ports
  login 269
post command 337
PostScript file
  converting from Diablo 630 file
    using ps630 command 408
  converting from Tektronix 4014 file
    using ps4014 command 407
  converting from troff
    using psroff command 414
  converting from troff intermediate file
    using psc command 410
    using psdit command 410
  reverse page order for printing
    using psrev command 414
  select page range for printing
    using psrev command 414
PostScript printers
  converting certain control characters for
    using piofquote command 301
power failure
  shutting down a system when there is a
    using rc.powerfail command 473
PPP 340
ppp subsystem
  startup and management
    pppcontrold daemon 340
PPP subsystem
  asynchronous connection 344
PPP subsystem *(continued)*
  asynchronous device stream attaching to
    using pppattachd daemon 338
  display RAS information
    pppstat command 345
pppattachd daemon 338
pppcontrold daemon 340
pppdial command 344
pppstat command 345
pr command 348
prctmp command 351
prdaily command 352
preprpnode command 352
prev command 355
Print Job Manager
  printing a file
    using pioout command 310
  processing a print job
    using piobe command 290
print queue
  prioritizing a job in the
    using qpri command 451
print queues
  creating new 307
printer
  burst pages
    pioburst command 292
  cancelling a job on the
    using qcan command 439
  creating a predefined data stream definition
    using piopredef command 312
  fonts
    copying 297
  providing the status of the spooling system
    using qstatus command 460
  scheduling jobs on the
    using qdaemon command 442
  starting a job on the
    using qprt command 452
printer backend
  send message to 309
printer backend commands
  piodmng 296
  piolsvp 301
  piomgpdev 304
  piomkapqd 305
  piomkpq 307
  piomsg 309
printer definition
  for virtual printer 295
printer definition file
  expanding and contracting 294
printer formatter
  driving
    using pioformat command 298
printer jobs
  enqueued by the enq command, scheduling
    using qdaemon command 442
printer queue
  displaying the status of a
    using qchk command 441

printer queue *(continued)*
removing from the system
using rmque command 595
using rmquedev command 596
printer spooling system
performing system administration for the
using qadm command 438
printer, virtual
removing
using rmvirprt command 620
printers
adding new 307
printf command 357
printing
holding a job 443
moving a spooled job 445
proccred command 360
processes
alters priorities of running
using renice command 515
showing status of
using ps command 391
procfiles command 363
procflags command 364
procldd command 366
procmap command 367
procrun command 368
procsig command 369
procstack command 370
procstop command 371
proctree command 372
procwait command 374
procwdx command 375
proff command 378
prompter command 379
prompting editor, invocation of, MH 379
proto command 380
prs command 383
prtacct command 386
prtconf 387
ps command 391
ps4014 command 407
ps630 command 408
psc command 410
psdit command 410
pseudo-device printers 304
pshare command 412
psplot command 413
psrev command 414
psroff command 414
pstart command 417
pstat command 418
ptf_pkg.$BLDCYCLE file
creating a PTF image
using ptfpkg command 420
ptfpkg command 420
ptx command 424
pwchange command 426
pwck command 428
pwdadm command 429
pwdck command 431
pwtokey command 434

Q

qadm command 438
qcan command 439
qchk command 441
qdaemon command 442
qhld command 443
qmov command 445
qpri command 451
qprt command 452
qstatus command 460
query program
starting Network Time Protocol
using ntpq command 212
queues
holding a spooled job 443
moving a spooled job 445
quot command 463
quota
consistency, checking file system
using quotacheck command 466
quota command 465
quotacheck command 466
quotaoff command 467
quotaon command 467
quotas
displaying disk usage and
using quota command 465
turning on and off file system
using quotaon or quotaoff command 467

R

RAM disk 597
RAM disk, virtual
removing
using rmramdisk command 597
ras_logger command 469
rc command 471
rc.mobip6 471
rc.powerfail command 473
rcp command 474
rcvdist command 478
rcvpack command 479
rcvstore command 479
rcvtty command 481
rdist command 481
rdistd command 494
rdump command 495
read command 497
reading
one line standard input
using read command 497
reboot command 499
recreatevg command 501
redefinevg command 503
reducevg command 503
refer command 505
refile command 507

refresh command 509
refrsrc command 510
remote command execution
providing server function for
using rshd daemon 648
remote machines' device
backing up files onto
using rdump command 495
remove command 515
removing
directories
using rmdir command 575
removing messages
from active status
using rmm command 585
remsh Command 645
renice command 515
reorgvg command 517
repl command 518
replacepv command 522
reports
latest installed maintenance level
using oslevel command 234
reset command 524
resize command 525
restart-secldapclntd 526
restbase command 527
restore Command 528
restricted shell
invoking
using Rsh command 643
restvg command 538
rev command 540
rexd daemon 542
rexec command 542
rexecd daemon 544
rgb command 545
RIPng and NDP daemon
for a router
using ndpd-router daemon 14
rlogin command 547
rlogind daemon 549
rm_niscachemgr command 553
rm_nisd command 554
rm_nispasswdd command 556
rmail command 557
rmaudrec command 557
rmC2admin command 561
rmCCadmin command 562
rmcctrl command 563
rmclass command 565
rmcomg command 566
rmcondition command 568
rmcondresp command 570
rmdel command 573
rmdev command 574
rmdir command 575
rmf command 577
rmfs command 578
rmgroup command 580
rmitab command 581
rmkeyserv command 581
rmlv command 583
rmlvcopy command 584
rmm command 585
rmnamsv command 586
rmnfs command 587
rmnfsexp command 588
rmnfsmnt command 589
rmpath command 591
rmprtsv command 593
rmps command 594
rmque command 595
rmquedev command 596
rmramdisk command 597
rmresponse command 598
rmrole command 600
rmrpdomain command 601
rmrpnode command 604
rmrset 605
rmrsrc command 606
rmsensor command 608
rmss command 611
rmuser command 617
rmvfs command 619
rmvirprt command 620
rmyp command 621
rndc 622
rndc-confgen command 623
roffbib command 624
route command 624
routed daemon 628
gateways file format 630
manipulating with SRC 631
signals 631
updating routing tables 629
router
NDP and RIPng daemon for
using ndpd-router daemon 14
routing tables, network
managing
using routed daemon 628
rpcgen command 637
rpcinfo command 638
rrestore Command 640
rsh command 645
Rsh command 643
rshd daemon 648
rstatd daemon 650
rt_enable command 651
runacct command 653
runcat command 656
rup command 656
ruptime command 657
ruser command 659
rusers command 660
rusersd daemon 661
rwall command 662
rwalld daemon 663
rwho command 664
rwhod command
manipulating with SRC 665

rwhod daemon 665

S

SCCS
files, displaying
using prs command 383
removing delta files
using rmdel command 573
SCCS commands
prs 383
rmdel 573
scripts
notifyevent 193
server function
remote command execution, providing
using rshd daemon 648
server function for rexec command, TCP/IP 544
service requests, handling
from PC-NFS clients
using pcnfsd daemon 635
shared login ports 412
shell variables
compared to standard input fields
using read command 497
shells
Bourne 643
restricted 643
simulating a system for
performance testing
using rmss command 611
SMIT
building printer dialogs 305
Creating print queues with 305
Creating printers with 305
source files 250
SRC
removing a subserver object definition 610
removing a subsystem notification method 590
removing a subsystem object definition 614
SRC configuration commands
rmnotify 590
rmserver 610
rmssys 614
standard input
reading one line
using read command 497
standard output
writing formatted output to
using printf command 357
startup
performing initialization for a normal
using rc command 471
status
of processes, showing
using ps command 391
subservers
removing SRC object definition 610
subsystem
requesting a refresh of
using refresh command 509
subsystems
removing a notification method 590
removing definition from SRC object class 614
system
removing a device from
using rmdev command 574
restarting
using reboot command 499
system boot
boot image
reading information from 527
system resource controller 590, 614
system shutdown
done when a power failure is detected
using rc.powerfail command 473
system tables
interpreting the contents of 418
System/370 Host Interface Adapter
checking for proper installation 244

T

table, NIS+
displaying
using niscat command 103
tables, network routing
managing
using routed daemon 628
tape devices
allowing remote access
rmt command 615
TCP/IP
commands
executing on a remote host 542
configuration database
controlling name server entries 9
configuration database managing entries
using ruser command 659
daemon
rexecd 544
daemons
named 1
hosts
listing logged in users 664
managing routing tables
using routed daemon 628
obtaining network status
using netstat command 30
parameters
tuning 154
print services
unconfiguring 593
providing server function
using rshd daemon 648
querying Internet domain name servers 198
routing tables
making manual entries 624
server function
providing 549
support for 665

TCP/IP *(continued)*
unconfiguring name services
using rmnamsv command 586
TCP/IP commands
namerslv 9
netstat 30
no 154
nslookup 198
rmnamsv 586
rmprtsv 593
route 624
ruser 659
rwho 664
TCP/IP daemons
rlogind 549
routed 628
rshd 648
rwhod 665
TCP/IP smit commands
namerslv 9
rmnamsv 586
rmprtsv 593
ruser 659
Tektronix 4014 file
converting to PostScript
using ps4014 command 407
termcap environment variable
setting to current window size
usng the resize Command 525
terminal settings
setting to current window size
usng the resize Command 525
terminals
initializing
using reset command 524
setting characteristics
using reset command 524
testing, performance
simulating a system for
using rmss command 611
text
changing format of
using newform command 40
translating
package format
using pkgtrans command 328
troff command
preprocessor for drawing figures on typesetter
using pic command 275
troff file
converting to PostScript
using psroff command 414
troff intermediate file format
converting to PostScript format
using psc command 410
using psdit command 410
tuning
network parameters
using no command 154

U

user
authenticate 243
changing a password
using passwd command 245
removing an account
using rmuser command 617
verifying password information about a
using pwdck command 431
users
administering passwords for
using pwdadm command 429
usingaix 515

V

virtual memory
displaying the system page size 242
virtual printer
displaying the attribute values of
using lsvirprt command 289
removing
using rmvirprt command 620
virtual printers 295, 301
virtual RAM disk
removing
using rmramdisk command 597
volume group
recreating a volume group 501
removing logical volumes
using rmlv command 583
removing physical volumes
using reducevg command 503
reorganizing the physical partition allocation
using reorgvg command 517

Readers' Comments — We'd Like to Hear from You

AIX 5L Version 5.2
Commands Reference, Volume 4

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

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.

Name _____ Address _____

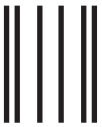
Company or Organization _____

Phone No. _____

Readers' Comments — We'd Like to Hear from You

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 H6DS-905-6C006
11501 Burnet Road
Austin, TX
78758-3493

Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold
Along Line