Bull

AIX Commands Reference Vol.4
named to rwhod

AIX

ORDER REFERENCE
86 A2 41JX 02

Bull

AIX Commands Reference Vol.4
named to rwhod

AIX

Software

April 2000

BULL ELECTRONICS ANGERS
CEDOC
34 Rue du Nid de Pie – BP 428
49004 ANGERS CEDEX 01
FRANCE

ORDER REFERENCE
86 A2 41JX 02

Printed in France

Suggestions and criticisms concerning the form, content, and presentation of this book are invited. A form is provided at the end of this book for this purpose.

To order additional copies of this book or other Bull Technical Publications, you are invited to use the Ordering Form also provided at the end of this book.

Trademarks and Acknowledgements

We acknowledge the right of proprietors of trademarks mentioned in this book.

AIX® is a registered trademark of International Business Machines Corporation, and is being used under licence.

UNIX is a registered trademark in the United States of America and other countries licensed exclusively through the Open Group.

Year 2000

The product documented in this manual is Year 2000 Ready.

The information in this document is subject to change without notice. Groupe Bull will not be liable for errors contained herein, or for incidental or consequential damages in connection with the use of this material.

Commands Reference, Volume 4....1
First Edition (October 1997)....1
Trademarks and Acknowledgements....5
About This Book....7
Alphabetical Listing of Commands....14
named Daemon....15
named4 Daemon....19
named8 Daemon....23
namerslv Command....27
ncheck Command....31
ndp Command....33
ndpd−host Daemon....35
ndpd−router Daemon....37
ndx Command....43
neqn Command....45
net Command....47
netlsd Daemon....55
netpmon Command....57
netstat Command....66
newaliases Command....78
newfile Command....79
newform Command....80
newgrp Command....83
newkey Command....85
news Command....87
next Command....89
nfsd Daemon....91
nfso Command....93
nfsstat Command....101
nice Command....105
nim Command....108
nimclient Command....121
nimconfig Command....125
nimdef Command....128
niminit Command....133
nis_cachemgr Daemon....137
nisaddcred Command....139
nisaddent Command....143
niscat Command....147
nischgrp Command....149
nischmod Command....151
nischown Command....153
nischttl Command....155
nisclient Command....158
nisdefaults Command....162
niserror Command....165
nisgrep Command....166
nisgrpadm Command....168
nisinit Command....170
nisln Command....174
nislog Command....175

nisls Command....177
nismatch Command....179
nismkdir Command....181
nismkuser Command....184
nisping Command....187
nispopulate Command....190
nisrm Command....194
nisrmdir Command....195
nisrmuser Command....197
nisserver Command....199
nissetup Command....202
nisshowcache Command....204
nisstat Command....205
nistbladm Command....207
nistest Command....212
nisupdkeys Command....214
nl Command....216
nm Command....219
no Command....222
nohup Command....234
nrglbd Daemon....236
nroff Command....238
nslookup Command....241
nsupdate Command....248
nsupdate8 Command....251
ntpdate Command....254
ntpq Command....257
ntptrace Command....264
ntsc Command....266
ntx_arp Command....268
ntx_chaddr Command....271
ntx_chinfo Command....274
ntx_chreset Command....277
ntx_comun Command....279
ntx_descr Command....282
ntx_dlli Command....284
ntx_dloop Command....287
ntx_ifcfg Command....289
ntx_load Command....291
ntx_log Command....293
ntx_macaddr Command....295
ntx_nms Command....297
ntx_nstat Command....299
ntx_ping Command....303
ntx_pserv Command....305
ntx_reset Command....307
ntx_route Command....309
ntx_snmp Command....311
ntx_stat Command....314
ntx_traps Command....316
ntx_ttype Command....319

ntx_ver Command....321
ntx_wdog Command....323
nulladm Command....325
number Command....326
od Command....327
odmadd Command....333
odmchange Command....335
odmcreate Command....336
odmdelete Command....339
odmdrop Command....340
odmget Command....341
odmshow Command....343
on Command....344
oslevel Command....346
ospf_monitor Command....348
pac Command....351
pack Command....353
packf Command....355
pagesize Command....357
panel20 Command....358
passwd Command....359
paste Command....363
patch Command....366
pathchk Command....372
pax Command....374
pcat Command....390
pclient Command....392
pcmciastat Command....394
pdelay Command....396
pdisable Command....398
penable Command....400
pg Command....402
phold Command....406
pic Command....408
pick Command....416
ping Command....420
pioattred Command....425
piobe Command....427
pioburst Command....430
piocnvt Command....432
piodigest Command....434
piodmgr Command....436
piofontin Command....438
pioformat Command....440
piofquote Command....443
piolpx Command....444
piolsvp Command....446
piomgpdev Command....450
piomkapqd Command....452
piomkpq Command....455
piomsg Command....458

piooout Command....460
piopredef Command....463
plotgbe Command....465
plotlbe Command....467
pmctrl Command....469
pop3d Daemon....476
portmap Daemon....478
portmir Command....480
post Command....483
pppattachd Daemon....485
pppcontrold Daemon....489
pppdial Command....493
pppstat Command....496
pprof Command....499
pr Command....501
prctmp Command....505
prdaily Command....506
prev Command....508
printenv Command....510
printf Command....511
prof Command....515
proff Command....518
prompter Command....520
proto Command....522
proxymngr Command....524
prs Command (SCCS)....526
prtacct Command....530
prtty Command....532
ps Command....533
ps630 Command....546
ps4014 Command....548
psc or psdit Command....550
pshare Command....553
psplot Command....555
psrev Command....556
psroff Command....557
pstart Command....561
pstat Command....563
ptx Command....565
pwd Command....567
pwdadm Command....568
pwdck Command....571

qadm Command....575
qcan Command....577
qchk Command....579
qdaemon Command....582
qhld Command....584
qmov Command....586
qosstat Command....588
qpri Command....590
qprt Command....592
qstatus Command....601
quiz Command....604
quot Command....607
quota Command....609
quotacheck Command....611
quotaon or quotaoff Command....613
ranlib Command....615
rc Command....617
rc.powerfail Command....618
rcp Command....620
rcvdist Command....624
rcvpack Command....626
rcvstore Command....627
rcvtty Command....629
rdist Command....631
rdump Command....638
read Command....641
readfile Command....644
readlvcopy Command....646
reboot or fastboot Command....647
redefinevg Command....649
reducevg Command....651
refer Command....653
refile Command....656
refresh Command....659
regcmp Command....661
rembak Command....663
remove Command....666
renice Command....667
reorgvg Command....669
repl Command....671
replacepv Command....676
repquota Command....678
reset Command....680
resize Command....682
resolve_links Command....684
restbase Command....686
restore Command....688
restvg Command....698
rev Command....700

revnetgroup Command....701
rexd Daemon....703
rexec Command....704
rexecd Daemon....706
rgb Command....708
ripquery Command....710
rlogin Command....712
rlogind Daemon....716
rm or delete Command....718
rm_niscachemgr Command....721
rm_nisd Daemon....723
rm_nispasswdd Daemon....725
rmail Command....727
rmclass Command....728
rmdel Command....729
rmdev Command....731
rmdir Command....734
rmdsmitd Command....736
rmf Command....738
rmfilt Command....740
rmfs Command....741
rmgroup Command....743
rmhist Command....745
rmitab Command....747
rmkeyserv Command....748
rmlv Command....750
rmlvcopy Command....752
rmm Command....754
rmnamsv Command....756
rmnfs Command....758
rmnfsexp Command....759
rmnfsmnt Command....761
rmnotify Command....763
rmprtsv Command....764
rmps Command....766
rmqos Command....768
rmque Command....769
rmquedev Command....771
rmrole Command....773
rmserver Command....775
rmss Command....777
rmssys Command....782
rmt Command....784
rmtun Command....786
rmuser Command....787
rmvfs Command....790
rmvirprt Command....791
rmyp Command....793
roffbib Command....795
route Command....797
routed Daemon....801

rpc.nisd Daemon....806
rpc.nispasswd Daemon....808
rpc.pcnfsd Daemon....810
rpcgen Command....813
rpcinfo Command....815
rpl Command....819
rrestore Command....821
rsh or remsh Command....825
rshd Daemon....829
rstatd Daemon....832
rtl_enable Command....833
runacct Command....836
runcat Command....840
rup Command....841
ruptime Command....843
ruser Command....845
rusers Command....848
rusersd Daemon....850
rwall Command....851
rwalld Daemon....853
rwho Command....854
rwhod Daemon....856

Commands Reference, Volume 4

First Edition (October 1997)

This edition of the *AIX Version 4.3 Commands Reference, Volume 4* applies to the AIX Version 4.3, 3270 Host Connection Program 2.1 and 1.3.3 for AIX, and Distributed SMIT 2.2 for AIX licensed programs, and to all subsequent releases of these products until otherwise indicated in new releases or technical newsletters.

The following paragraph does not apply to the United Kingdom or any country where such provisions are inconsistent with local law: THIS MANUAL IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND 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.

It is not warranted that the contents of this publication or the accompanying source code examples, whether individually or as one or more groups, will meet your requirements or that the publication or the accompanying source code examples are error−free.

This publication 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.

It is possible that this publication may contain references to, or information about, products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that such products, programming, or services will be offered in your country. Any reference to a licensed program in this publication is not intended to state or imply that you can use only that licensed program. You can use any functionally equivalent program instead.

The information provided regarding publications by other vendors does not constitute an expressed or implied recommendation or endorsement of any particular product, service, company or technology, but is intended simply as an information guide that will give a better understanding of the options available to you. The fact that a publication or company does not appear in this book does not imply that it is inferior to those listed. The providers of this book take no responsibility whatsoever with regard to the selection, performance, or use of the publications listed herein.

NO WARRANTIES OF ANY KIND ARE MADE WITH RESPECT TO THE CONTENTS, COMPLETENESS, OR ACCURACY OF THE PUBLICATIONS LISTED HEREIN. ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE SPECIFICALLY DISCLAIMED. This disclaimer does not apply to the United Kingdom or elsewhere if inconsistent with local law.

A reader's comment form is provided at the back of this publication. If the form has been removed, address comments to Publications Department, Internal Zip 9561, 11400 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.

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON−INFRINGEMENT.

THIS PUBLICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THE PUBLICATION. HEWLETT−PACKARD COMPANY, INTERNATIONAL BUSINESS MACHINES CORP., SUN MICROSYSTEMS, INC., AND UNIX SYSTEMS LABORATORIES, INC., MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS PUBLICATION AT ANY TIME.

This software and documentation is based in part on the Fourth Berkeley Software Distribution under license from The Regents of the University of California. We acknowledge the following institutions for their role in its development: the Electrical Engineering and Computer Sciences Department at the Berkeley Campus.

The Rand MH Message Handling System was developed by the Rand Corporation and the University of California.

Portions of the code and documentation described in this book were derived from code and documentation developed under the auspices of the Regents of the University of California and have been acquired and modified under the provisions that the following copyright notice and

Trademarks and Acknowledgements

The following trademarks and acknowledgements apply to this book:

ADM is a trademark of Lear Siegler, Inc.

AIX is a registered trademark of International Business Machines Corporation.

Connect is a trademark of INTERACTIVE Systems Corporation.

DEC is a trademark of Digital Equipment Corporation.

DEC VT100, VT220, VT320, and VT330 are trademarks of Digital Equipment Corporation.

GL is a trademark of Silicon Graphics, Inc.

HP is a trademark of Hewlett−Packard Company.

IBM is a registered trademark of International Business Machines Corporation.

INed is a trademark of INTERACTIVE Systems Corporation.

InfoExplorer is a trademark of International Business Machines Corporation.

Intel is a trademark of Intel Corporation.

Interleaf is a trademark of Interleaf, Inc.

LaserJet Series II is a trademark of Hewlett−Packard Company.

Micro Channel is a registered trademark of International Business Machines Corporation.

NetView is a trademark of International Business Machines Corporation.

Network Computing System is a trademark of Apollo Computer, Inc.

OSF and OSF/Motif are trademarks of Open Software Foundation, Inc.

Personal Computer AT and AT is a registered trademark of International Business Machines Corporation.

Personal System/2 is a registered trademark of International Business Machines Corporation.

PS/2 is a registered trademark of International Business Machines Corporation.

POSIX is a trademark of the Institute of Electrical and Electronic Engineers (IEEE).

PostScript is a trademark of Adobe Systems Incorporated.

Proprinter is a registered trademark of International Business Machines Corporation.

Quickwriter is a registered trademark of International Business Machines Corporation.

Quiet is a trademark of International Business Machines Corporation.

RS/6000 is a trademark of International Business Machines Corporation.

RT is a registered trademark of International Business Machines Corporation.

Sun is a trademark of Sun Microsystems, Inc.

Tektronix is a trademark of Tektronix, Inc.

Televideo is a trademark of Televideo, Inc.

The Source is a service mark of Source Telecomputing Corp., a subsidiary of The Reader's Digest Assn., Inc.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited.

WY−50 is a trademark of the WYSE Corporation.

WYSE is a trademark of WYSE Corporation.

About This Book

This book is Volume 4 of the six−volume *AIX Version 4.3 Commands Reference*, SBOF−1877, which contains reference information on Advanced Interactive Executive (AIX) 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 AIX 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 AIX 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 diagram showing command line options.

Description A discussion of the command describing in detail its function and use.

Flags A list of command line flags and associated variables with an explanation of how the flags modify the action of the command.

Parameters A list of command line parameters and their descriptions.

Subcommands A list of subcommands (for interactive commands) that explains their use.

Exit Status A description of the exit values the command returns.

Security Specifies any permissions needed to run the command.

Examples Specific examples of how you can use the command.

Files A list of files used by the command.

Related Information A list of related commands in this book and related discussions in other books.

Implementation Specifics

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
```

Syntax Diagrams

AIX command syntax is represented by syntax diagrams and usage statements.

Syntax diagrams are designed to provide information about how to enter the command on the command line. A syntax diagram can tell you:

- Which flags can be entered on the command line
- Which flags must take a parameter
- Which flags have optional parameters
- Default values of flags and parameters, if any
- Which flags can and cannot be entered together
- Which flags and parameters are optional

- When you can repeat flag and parameter sequences.

AIX commands use the following conventions in their syntax diagrams:

- Diagram items that must be entered literally on the command line are in **bold**. These items include the command name, flags, and literal characters.
- Diagram 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*.
- Default values that do not have to be entered are in the normal font on a **bold** path.

The Sample Syntax Diagram illustrates the conventions used in syntax diagrams. Each part of the diagram is labeled. An explanation of the labels follows the diagram.

You interpret the example diagram as follows.

0 PATH LINE The path line begins the syntax diagram.

1 COMMAND NAME This item in the diagram is the name of the command you want to invoke. It is in bold, which indicates that it must be entered exactly as it appears in the diagram.

In the example diagram, the path branches into two paths after the command name. You can follow either the lower path (discussed in item 2) or the upper path (discussed in item 3).

2 SINGLE CHOICE BOX If you follow the lower path, you encounter a box with the words *one of* over it. You can choose only one item from this box.

3 DEFAULT LINE If you follow the upper path, you bypass the single choice box, and enter nothing. The bold line around the box is a default line, which means that you do not have to enter anything from that part of the diagram. Exceptions are usually explained under "Description." One important exception, the blank default line around input and output files, is explained in item 10.

4 REPEAT ARROW When you follow a path that takes you to a box with an arrow around it, you must choose at least one item from the box. Then you can either follow the arrow back around and continue to choose items from the box, or you can continue along the path. When following an arrow that goes around a box (rather than an arrow that includes several branches in the diagram), do not choose the same item more than once.

5 REQUIRED ITEM Following the branch with the repeat arrow is a branch with three choices and no default line around them. This means that you must choose one of A, B, or C.

6 GO TO NEXT LINE If a diagram is too long to fit on one line, this character tells you to go to the next line of the diagram to continue entering your command. Remember, the diagram does not end until you reach the vertical mark.

7 CONTINUE DIAGRAM This character shows you where to continue with the diagram after it breaks on the previous line.

8 OPTIONAL PARAMETER If a flag can (but does not have to) take a parameter, the path branches after the flag. If you cannot enter a space between the flag and parameter, you are told in a footnote.

9 DEFAULT VALUE Often, a command has default values or actions that it will follow if you do not enter a specific item. These default values are indicated in normal font in the default line if they are equivalent to something you could enter on the command line (for example, a flag with a value). If the default is not something you can enter on the command line, it is not indicated in the diagram.

> **Note:** Default values are included in the diagram for your information. It is not necessary to enter them on the command line.

10 INPUT OR OUTPUT A command that can read either input files or standard input has an empty default line above the file parameter. If the command can write its output to either an output file or to standard output, it is also shown with an empty default line above the output file parameter.

If a command can read only from standard input, an input file is not shown in the diagram, and standard input is assumed. If a command writes only to standard output, an output file is not shown in the diagram, and standard output is assumed.

When you must supply a file name for input or output, the file parameter is included in the diagram without an empty default line above it.

11 FOOTNOTE If a command has special requirements or restrictions, a footnote calls attention to these differences.

12 VERTICAL MARK This ends the syntax diagram.

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 & (ampersand) 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

When you work with AIX, 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 AIX 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 in AIX, 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.

AIX 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.

AIX 32−Bit Support for the X/Open UNIX95 Specification

Beginning with AIX Version 4.2, the operating system is designed to support the X/Open UNIX95 Specification for portability of UNIX−based operating systems. Many new interfaces, and some current ones, have been added or enhanced to meet this specification. Beginning with Version 4.2, AIX is even more open and portable for applications.

At the same time, compatibility with previous AIX releases 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 UNIX95−portable application, you may need to refer to the X/Open UNIX95 Specification, which can be obtained on a CD−ROM by ordering the printed copy of *AIX Version 4.3 Commands Reference*, order number SBOF−1877, or by ordering *Go Solo: How to Implement and Go Solo with the Single Unix Specification*, order number SR28−5705, a book which includes the X/Open UNIX95 Specification on a CD−ROM.

AIX 32−Bit and 64−Bit Support for the UNIX98 Specification

Beginning with AIX Version 4.3, the operating system is designed to support the X/Open 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 AIX Version 4.3 even more open and portable for applications.

At the same time, compatibility with previous AIX releases 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 X/Open UNIX98 Specification, which can be obtained on a CD−ROM by ordering the printed copy of *AIX Version 4.3 Commands Reference*, order number SBOF−1877, or by ordering *Go Solo: How to Implement and Go Solo with the Single Unix Specification*, order number SR28−5705, a book which includes the X/Open UNIX98 Specification on a CD−ROM.

Related Information

The following books contain information about or related to commands:

- *AIX and Related Products Documentation Overview*, Order Number SC23−2456.
- *AIX Version 4.3 Files Reference*, Order Number SC23−4168.
- *AIX Version 4.3 General Programming Concepts: Writing and Debugging Programs*, Order Number SC23−4128.
- *AIX Version 4.3 Problem Solving Guide and Reference*, Order Number SC23−4123.
- *AIX Version 4.3 System Management Guide: Communications and Networks*, Order Number SC23−4127.
- *AIX Version 4.3 System Management Guide: Operating System and Devices*, Order Number SC23−4126.
- *AIX Version 4.3 System User's Guide: Operating System and Devices*, Order Number SC23−4121.
- *AIX Version 4.3 System User's Guide: Communications and Networks*, Order Number SC23−4122.
- *AIX Versions 3.2 and 4 Performance Tuning Guide*, Order Number SC23−2365.
- *AIX Version 4.3 Guide to Printers and Printing*, Order Number SC23−4130.
- *AIX Version 4.3 Kernel Extensions and Device Support Programming Concepts*, Order

Number SC23−4125.

- *5080 Graphics System Installation, Operation, and Problem Determination*, Order Number GA23−2063.
- *AIX Version 4.3 Technical Reference: Base Operating System and Extensions Volume 1* Order Number SC23−4159
- *AIX Version 4.3 Technical Reference: Base Operating System and Extensions Volume 2*, Order Number SC23−4160.
- *AIX Version 4.3 Technical Reference: Communications Volume 1*, Order Number SC23−4161.
- *AIX Version 4.3 Technical Reference: Communications Volume 2*, Order Number SC23−4162.
- *AIX Version 4.3 Technical Reference: Kernel and Subsystems Volume 1*, Order Number SC23−4163.
- *AIX Version 4.3 Technical Reference: Kernel and Subsystems Volume 2*, Order Number SC23−4164.
- *AIX Version 4 Keyboard Technical Reference*, Order Number SC23−2631.
- *Distributed SMIT 2.2 for AIX: Guide and Reference*, Order Number SC23−2667.
- *3270 Host Connection Program 2.1 and 1.3.3 for AIX: Guide and Reference*, Order Number SC23−2563.

The following books also may be helpful:

- Lamb, Linda. *Learning the vi Editor*. Sebastopol, CA: O'Reilly & Associates, 1990. Order Number SR28−4966.
- Dougherty, Dale. *sed & awk*. Sebastopol, CA: O'Reilly & Associates, 1990. Order Number SR28−4968.
- Hunt, Craig. *TCP/IP Network Administration*. Sebastopol, CA: O'Reilly & Associates, 1992. Order Number SR23−7422.

Ordering Publications

You can order publications from your sales representative or from your point of sale.

To order additional copies of this book, use order number SC23−2540.

To order additional copies of all six volumes of *AIX Version 4.3 Commands Reference*, use Order Number SBOF−1877.

Use *AIX and Related Products Documentation Overview* for information on related publications and how to obtain them.

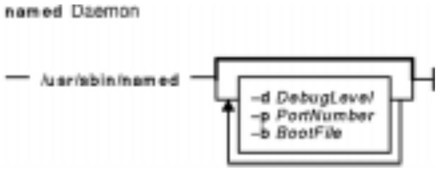
Alphabetical Listing of Commands

named Daemon

Purpose

Provides the server function for the Domain Name Protocol.

Syntax



/usr/sbin/named [**−d** *DebugLevel*] [**−p** *PortNumber*] [**−b** *BootFile*]

Description

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

> **Note:** The **named** 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 **named** 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 **named** 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 **named** daemon does not function properly.

Manipulating the named Daemon with the System Resource Controller

The **named** daemon is a subsystem controlled by the System Resource Controller (SRC). The **named** 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 **named** 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 **named** 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 **named** daemon listens for DOMAIN requests. If this variable is not specified, the **named** 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 **named** daemon process using the **kill** command:

SIGHUP The **named** 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 **named** 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 **named** 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 **named** daemon turns off debugging.

SIGABRT/SIGIOT The **named** daemon dumps the current statistics to the **/var/tmp/named.stat** file.

Examples

1. To start the **named** 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 **named** daemon is stored in the **/etc/named.pid** file upon startup.

2. To stop the **named** 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 **named** 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 **named** daemon, enter:

   ```
   traceson −s named
   ```

   OR

   ```
   kill −30 'cat /etc/named.pid'
   ```

   The **named** 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 **named** daemon, enter:

   ```
   tracesoff
   ```

   OR

   ```
   kill −35 'cat /etc/named.pid'
   ```

   Either of these commands immediately turns off all debugging.

6. To start the **named** 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.

7. To view the contents of the DHCP Server database files **/etc/dhcpsd.ar** and **/etc/dhcpsd.cr**, enter:

   ```
   lssrc −l −s dhcpsd
   ```

Files

/usr/sbin/named 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 **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 Version 4.3 System Management Concepts: Operating System and Devices*.

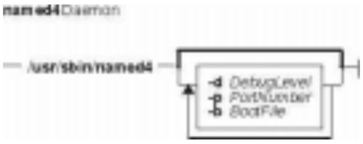
Configuring Name Servers and Planning for DOMAIN Name Resolution in *AIX Version 4.3 System Management Concepts: Operating System and Devices*.

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.

AIX 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 Version 4.3 System Management Guide: Communications and Networks*.

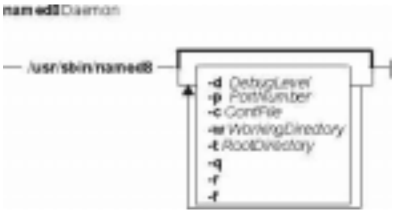
Configuring Name Servers and Planning for DOMAIN Name Resolution in *AIX Version 4.3 System Management Guide: Communications and Networks*.

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.

AIX 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 visa−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.

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.

SIGABRT/SIGIOT The**named8** daemon dumps the current statistics to the **/var/tmp/named.stat** file.

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 Version 4.3 System Management Guide: Communications and Networks*.

Configuring Name Servers and Planning for DOMAIN Name Resolution in *AIX Version 4.3 System Management Guide: Communications and Networks*.

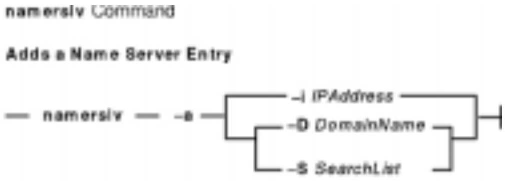
namerslv Command

Purpose

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

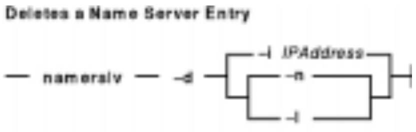
Syntax

To Add a Name Server Entry



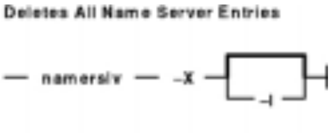
namerslv −a {**−i***IPAddress* | **−D***DomainName*| **−S***SearchList*}

To Delete a Name Server Entry



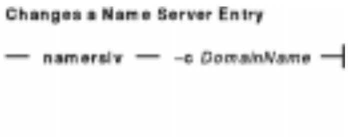
namerslv −d {**−i***IPAddress* | **−n** | **−l**}

To Delete All Name Server Entries



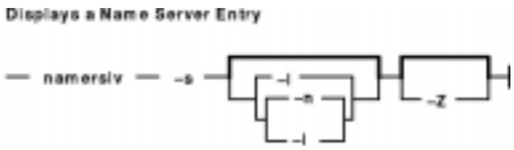
namerslv −X [**−I**]

To Change a Name Server Entry



namerslv −c *DomainName*

To Display a Name Server Entry



namerslv−s [**−I** | **−n** | **−l**] [**−Z**]

To Create the Configuration Database File

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

To Rename the Configuration Database File

namerslv −E *FileName*

To Move the Configuration Database File to Prevent Name Server Use

namerslv −e

To Import a File into the Configuration Database File

namerslv −B *FileName*

To Change a Search List Entry

namerslv −C *Search List*

Description

The **namerslv** low−level command adds or deletes domain name server entries for local resolver routines in the system configuration database. By default, the system configuration database is contained in the **/etc/resolv.con f** 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.con f** 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`, enter:

   ```
   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`, enter:

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

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

   ```
   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, enter:

   ```
   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`, enter:

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

Files

/usr/sbin/namerslv	Contains the **namerslv** command.
/etc/resolv.con f	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 Version 4.3 System Management Guide: Communications and Networks*.

Setting up and running Web−based System Manager in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

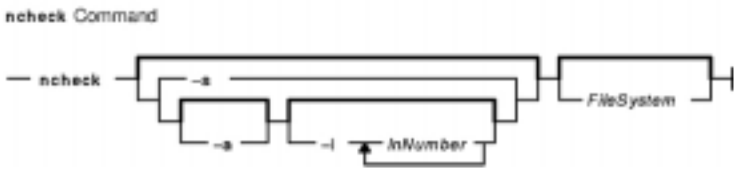
The TCP/IP Reference in *AIX Version 4.3 System Management Guide: Communications and Networks*.

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.

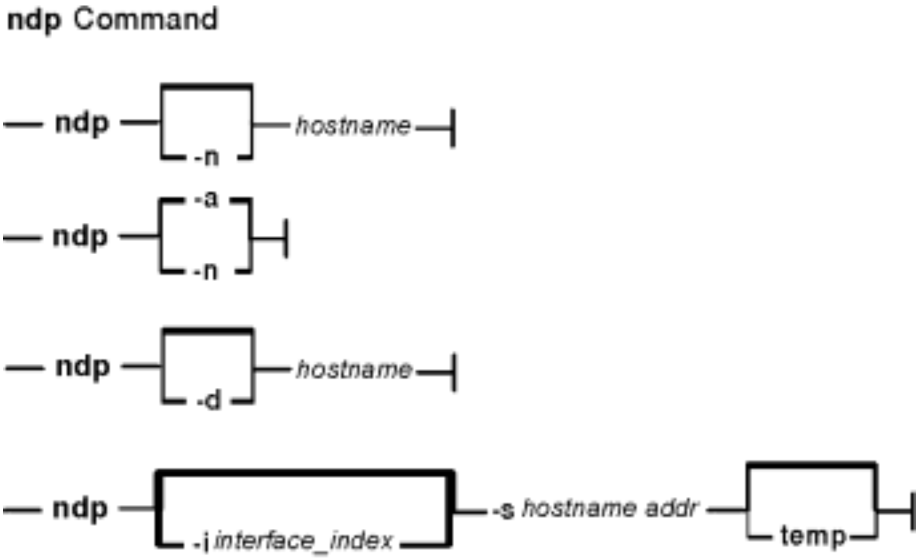
The File Systems Overview for System Management in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

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

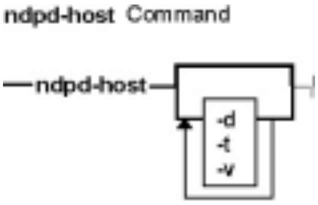
The **ifconfig** command, **ndpd−host** command, **ndpd−router** command, and the **autoconf6** command.

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

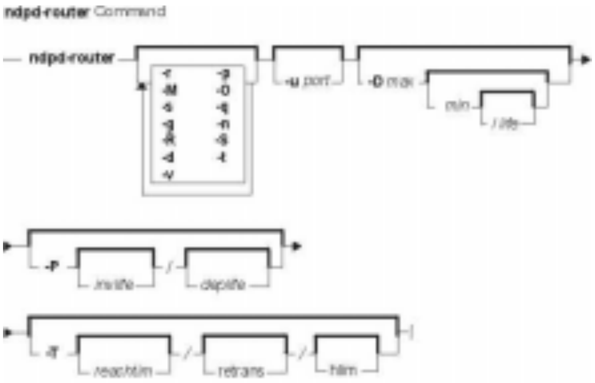
The **ifconfig** command, **route** command, **autoconf6** command, and the **ndpd−router** command.

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**] [**−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 provides 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, separated by spaces.

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.

In order to send Router Advertisements and/or RIPng packets, both local and remote Link−Local addresses must be configured.

Flags

−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 Set the **O** flag (other stateful information) in advertisements

−p Do 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 DO 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 Do 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 Do 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

```
option [option−directive ...]
Set different per−interface options.
```

The possible *option−directive* are:

mtu [**=***mtuval*] Advertise 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**) Do not listen (listen) to incoming rip packets. Do not send (send) rip 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*]) Do not send (send) router advertisements. With *min*[**/***max*] option, set the interval between router advertisements.

flag = [**MO**] Sets the stateful mode flags in router advertisements.

M use stateful configuration

O use stateful configuration, but not for addresses

life =*dd* Sets the router life field in router advertisements (value in seconds).

reach =*dd*

Set the reachable field in router advertisements (value in seconds).

retrans =*dd*

Set the retransmit interval field in router advertisements (value in 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. list is comma−separated. One can use le* to match all the leX interfaces. The default option must be the first line in the **/etc/gateway6** file.

filter = (**in**|**out**) [*filter−directive* ...]

Define a filter pattern for incoming (**filter=in**) or outgoing (**filter=out**) rip 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 rip 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 rip 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/*dd*. The last action means: if the pattern matches and if prefix length is greater or equal to *dd*, accept the prefix with new length *dd*. 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 explicit outgoing filter:

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

The possible filter−directive are:

prefix =*xxx****::/****dd*

The pattern matches only if *xxx***::/***dd* is a prefix of the prefix in the rip packet.

gateway =*xxx*

The pattern matches only if the rip message comes from source address *xxx*, only in incoming filters.

length = (=|>=|<=|<|>)*dd*

The pattern match only if the prefix length in the rip message is equal (greater, less, ..) to *dd*.

action = (accept|reject|truncate/*dd)*

Specify the action to perform if the pattern matches: accept the message, reject the message, accept but truncate the prefix to dd bits.

if =*list*
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. You can use `le*` to specify all the leX interfaces.

prefix [*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.

The possible *prefix−directive* are:

prefix =*xxx****::/****dd*

The advertised prefix.

flag = [LA]

Set the **L** and/or **A** flag for the prefix (the default is **LA**).

deprec =*dd*

Set the deprecated time for the prefix (value in seconds).

valid =*dd*

Set the validity time for the prefix (value in seconds).

if =*xx*
interface =*xx*

Specifies the interface on which the directive applies (mandatory).

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

The syntax is as follows:

```
xxx::/dd metric dd
xxx::/dd metric dd gateway zzz ifname
```

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

/etc/gateway6 Examples

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 **ctiXX** interfaces.

Note: 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 rip 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, **route** command, **autoconf6** command, **ndpd−host** command.

ndx Command

Purpose

Creates a subject−page index for a document.

Syntax

ndx Command
ndx
SubjectFile
"FormatterCommandLine "

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

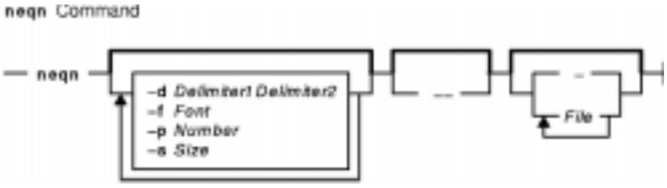
The **mm** command, **mmt** command, **nroff** command, **subj** command, **troff** command.

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.

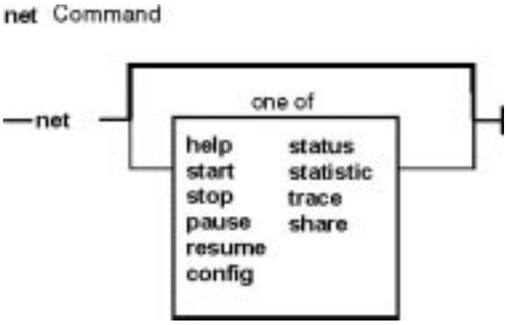
The **eqnchar** file format.

net Command

Purpose

Configures and controls Fast Connect servers.

Syntax



net [**help** | **start** | **stop** | **pause** | **resume** | **config** | **status** | **statistics** | **trace** | **share**]

Description

The **net** command configures and controls Fast Connect servers.

Subcommands

help	Displays help on the subcommand.
start	Starts the server.
stop	Stops the server.
pause	Stops the server temporarily.
resume	Resumes the paused server.
config	Lists and changes configuration parameters for the server.
status	Gives status of the server.
statistics	Gives statistics on server resources.
trace	Turns the server tracing on and off.
share	Lists, adds or deletes file and printer shares on the server.

net help Subcommand

Syntax

net help*subcommand*

or

net*subcommand***help**

Description

Provides help information about the *subcommand*s.

net start Subcommand

Purpose

Starts the server

Syntax

net start [**/load**]

Description

The **start** subcommand starts and initializes the server using parameters from the configuration file.

It can start the server only if the server process is already loaded but the server is in stopped (not running) state. The server can be in this state either in the beginning when the process was started at system initialization time or after stopping the server using **net stop**.

Flags

/load Loads the server process if it is not already loaded.

Return Codes

0 The server (%s) is already running.

0 The server (%s) has started successfully.

1 Syntax error was detected: Unknown keyword or command option (%s).

2 The server (%s) could not be started because its process was not running.

3 The request is not valid for the current state of the server (%s).

4 Operation could not be performed.

net stop Subcommand

Purpose

Stops and terminates the server.

Syntax

net stop [**/unload**]

Description

The **net stop** subcommand stops and unloads the server. It can stop the server only if it is running or paused. Once stopped, the server can be restarted using **net start**.

Flags

/unload Unloads the server process.

Return Codes

0 The server (%s) has stopped successfully.

0 The server (%s) has stopped and its process unloaded successfully.

1 Syntax error detected: Unknown keyword or command option (%s).

2 The request is not valid for the current state of the server (%s).

3 Error in unloading the server process on the server (%s).

4 Operation could not be performed.

5 Either cifsPrintServer is not running or it could not be terminated.

net pause Subcommand

Purpose

Pauses the server

Syntax

net pause

Description

The **net pause** subcommand pauses the server. It can pause the server only if it is running. Once paused, the server does not accept any new connections but continues serving the existing ones. It can be resumed with **net resume**.

Return Codes

0 The server (%s) has paused successfully.

1 Syntax error detected: Unknown keyword or command option (%s).

2 The request is not valid for the current state of server

3 Operation could not be performed.

net resume Subcommand

Purpose

Resumes the server.

Syntax

net resume

Description

The **net resume** subcommand resumes the server. It can resume the server only if it is paused. Once resumed, it starts accepting new connections.

Return Codes

0 The server (%s) has resumed successfully.

1 Syntax error detected: Unknown keyword or command option (%s).

2 The request is not valid for the current state of server

3 Operation could not be performed.

net config Subcommand

Purpose

Lists and changes the configuration parameters of the server.

Syntax1

net config

Syntax2

net config /component

Syntax3

net config /component:*cname***/***parameter***:***val*

Syntax4

net config /listparm /component:*cname***/parameter:***parameter*

Description

The **net config** subcommand lists and changes the configuration parameters of the server. For example:

Syntax1 Lists the configuration parameters.

Syntax2 Lists all the components or groups of configuration parameters for the server.

Syntax3 Adds or changes the given *parameter* for the given component *cname*.

Syntax4 Lists the entry for the given *parameter* for the given component *cname* from the configuration file.

Note: The configuration parameter can only be changed by the root user.

Flags

/listparm Lists the given parameter for the given component.

/component:*cname* Specifies the component in the configuration file whose parameter needs to be added or changed. Valid values for *cname* are **smbserver**, and one of the shares specified for *shares* parameter of the smbserver. The default component is **smbserver**.

The *parameter* can be one of the following:

/maxconnections:*number*
: Maximum number of connections to server resources. 0 specifies unlimited number.

/maxusers:*number*
: Maximum number of users (sessions) that are permitted. 0 specifies unlimited number.

/autodisconnect:*number*
: Timeout (in minutes) for inactive, unused sessions.

/maxopens:*number*
: Maximum number of open files on the server. 0 specifies unlimited number.

/maxsearches:*number*
: Maximum number of open searches on the server. 0 specifies unlimited number.

/servername:*s_name*
: The name of the server.

/domainname:*d_name*
: The name of the domain, server belongs to.

/guestname:*g_name*
: Logon name as guest on the server.

/passthrough_authentication_server:*pas_name*
: The name of the passthrough authentication server.

/backup_passthrough_authentication_server:*bpas_name*
: The name of the backup passthrough authentication server.

/primary_wins_ipaddr:*pwins_addr*
: Specifies the dotted IP address of the primary WINS server.

/secondary_wins_ipaddr:*swin_ipaddr*
: Specifies the dotted IP address of the secondary WINS server.

/wins_proxy:0|1
: Specifies whether or not the server has to act as WINS PROXY. Valid values are 0 for no and 1 for yes with 0 as the default.

/send_file_api:0|1
: Specifies whether or not the **send_file API** is to be used. Valid values are 0 for off and 1 for on with 1 as the default.

/send_file_size:*sf_size*
: If the **send_file_api** is 1 and the requested SMB read size is greater than the value of this parameter, **send_file API** will be used in the SMB operation. The value ranges between 1 and 4194304 with 4096 as the default value.

/send_file_cache_size:*sfc_size*
: If the **send_file_api** is 1 and the requested SMB read size is less than the value of this parameter, the **send_file API** will cache the file. The value ranges between 0 and 4194304 with 0 as the default value which means that the **send_file API** will not cache the file.

/umask:*u_mask*
: AIX **umask**. It is an octal value and ranges between 0 and 0777 with 022 as the default.

/guestlogonsupport: 0|1
: Specifies whether or not guest access is allowed. Valid values

are 0 for no and 1 for yes with 0 being the default.

/dosattrmapping:0**|**1

If set to 1, then Archive, System, and Hidden attributes will be mapped to user, group, and other execute bits respectively. Otherwise, these attributes are not supported.

/dosfilenamemapping:0**|**1

If set to 1, long filenames will be mapped to 8.3 format. Otherwise long filenames will be truncated.

/dosfilenamemapchar:m_char*

The characters used to map long filenames to 8.3 format. Valid values are'~' and '^' with '~' being the default.

Return Codes

0 Command completed successfully.

1 Syntax error: Unknown keyword or command option (%s).

2 Command could not be executed. Invalid parameter value (%s).

3 Operation could not be performed.

Output for syntax1 command ***net config***

```
Server Name                             ........
Server Description                      ........
Domain Name                             ........
AIX Fast Connect Software version       ........
Guest ID Name                           ........
```

Assuming that the smbserver had entries:

```
adapters = tr0 tr1 and
shares   = file0 print1)
```

Output for syntax2 command ***net config /component***

```
smbserver
tr0
tr1
file0
print1
```

Assuming that the smbserver had entries

```
adapters = tr0 tr1 and
shares   = file0 print1
```

Output for syntax4 command ***net config /listparm /component:smbserver /parameter: adapters***

```
tr0 tr1
```

Output for syntax4 command ***net config /listparm /component:smbserver /parameter: shares***

```
file0 print1
```

net status Subcommand

Purpose

Displays status of the server.

Syntax

net status

Description

The **status** subcommand displays status of the server. It tells whether the status of the server is running, paused, or stopped.

Return Codes

0 Server (%s) is running.

1 Syntax error: Unknown keyword or command option (%s).

2 Server (%s) is not running.

3 Server (%s) has been paused.

4 Operation could not be performed

net statistics Subcommand

Purpose

Displays the statistics on server resource usage.

Syntax

net statistics [**/reset**]

Description

Lists the statistics on server resources since it was started or resets the statistics.

Flags

/reset Resets all statistic fields for the server.

Return Codes

0 Command completed successfully.

1 Syntax error: Unknown keyword or command option (%s).

2 Operation could not be performed.

Output

Server statistics for server (%s) since %s time

```
Sessions started              .....
Sessions timedout             .....
Sessions dropped              .....
Password Errors               .....
Permission Errors             .....
System Errors                 .....
Bytes sent low                .....
Bytes sent high               .....
Bytes received low            .....
Bytes received high           .....
Average response              .....
Request buffer failures       .....
Big buffer failures           .....
Jobs  queued                  .....
```

net trace Subcommand

Purpose

Turns tracing on or off for the server.

Syntax1

net trace /on

Syntax2

net trace /off

Description

Turns tracing on or off for the server. The user does not have to start or stop the server.

Flags

/on Turns tracing on.

/off Turns tracing off.

Return Codes

0 Command completed successfully.

1 Syntax error: Unknown keyword or command option (%s).

2 Operation could not be performed.

net share Subcommand

Purpose

To list, add and delete file or printer shares on the server.

Syntax1

net share [**/infolevel:0**|**1**|**2** | **/netname:***share_name*]

Syntax2

net share /add /netname:*share_name***/type:***file***/path:***path_name* [**/desc:***share_desc*]

Syntax3

net share /add /netname:*share_name***/type:***printer***/printq:***qname* [**/print_options:***ostr* | **/desc:***share_desc*]

Syntax4

net share /delete /netname:*share_name*

Description

Syntax1 lists one or more shares.
Syntax2 adds a file share to the server.
Syntax3 adds a printer share.
Syntax4 deletes a share from the server.

Note: To change a share, you must first delete it and then add it again.

Flags

/add Adds a share to the server.

/deletes Deletes a share from the server.

/infolevel Specifies the level of information desired. Default level is 1. Valid values are 0, 1 and 2 and 99.

/type Specifies the share type. Valid values are **file** and **printer**. Default value is **file**.

/netname:*share_name* Name of the share that is to be listed, added or deleted. Without this option, all shares will be listed.

/path:*path_name* Absolute pathname of the share which must be specified if it is of type **file**.

/printq:*qname* Name of the share queue which must be specified if it is of type **printer**.

/print_options:*ostr* String specifying printer options.

/desc:*desc* Brief description of the share.

Return Codes

0 Command completed successfully.

1 Syntax error: Unknown keyword or command option (%s).

2 Operation could not be performed.

3 Command could not be executed. Invalid value (%s) of parameter.

4 Syntax Error: The share path or queue name must be specified.

5 Error adding the share − share name already exists.

6 Error deleting the share − share name not found.

7 The configuration file could not be updated to reflect the current change.

Output for info level 0 :

```
netname1
netname2
...
netnamen
```

Output for info level 1 :

```
Share Name      Share Type      Path Name or        Share Description
                                Queue Name
-----------     ----------      ---------------     ------------------

netname1        File            /home/name/xxx      File Description
netname2        Printer         lpq1                Printer Description
......          ......          ....                ..........
netnamen        Printer         lpq2                Printer Description
```

Output for info level 99 :

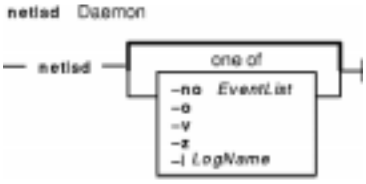
```
netname:%s:type:%s:path:%s:printq:%s:print_options:%s:desc:%s::
```

netlsd Daemon

Purpose

Starts the license server.

Syntax



netlsd [**−no** *EventList* | **−o** | **−v** | **−z** | **−l** *LogName*]

Description

The **netlsd** command starts a license server on the local node. There is no graphical interface for this command.

Flags

−l*LogName* (Lowercase L) Overrides the default name and location of the file used to store log information. This allows the I/O activity to the files used by the license server to be spread across multiple disks which can be important for large installations.

−no Turns off logging of the events specified in *EventList*. Any combination of events is valid, but items in the list of events must not be separated by spaces or other characters. The event types that you can specify are the following:

c
License check−in events. Licensed products usually check in with the license server at regular intervals while you are using the product.

e
Error events.

l
License−grant and license−release events.

m
Message events.

p
Product events: a product was added, renamed, or deleted.

s
License server start/stop events.

t
License timeout events. When a licensed product fails to check in with the license server, it may stop running after it "times out." The vendor of the product sets the timeout interval, which is how long a product may run after it has lost contact with the license server.

v
Vendor events: a vendor was added, renamed, or deleted.

w

Waiting events. These include:

wait events
: You are waiting for a license.

wait−grant events
: You are waiting for and then are granted a license.

wait−remove events
: You are waiting for a license and then asked to be removed from the queues before a license was granted.

−o Overrides the in−use flag at a license server database. While a license server is running, its database is flagged as being in use to prevent more than one server from running on the same node. When a license server stops running, the flag is reset. However, if a license server exits abnormally, the flag may not be reset, which prevents the server from restarting. This option overrides the in−use flag and allows the server to be restarted. Do not use **−o** unless you are sure the license server is not running.

−v NetLS library verbose mode.

−z Debugging flag. (Prints RPC debugging information.)

Examples

1. To start a license server; do not log check−in, vendor, product, timeout, or message events, enter:

   ```
   netlsd −no cvptm
   ```

2. To start a license server, overriding the in−use flag, enter:

   ```
   netlsd −o
   ```

3. To start a license server, overriding the default log file, enter:

   ```
   netlsd −l /logs/license_server_log
   ```

Files

lic_db	Contains vendor, product and license details. (Encrypted binary file)
lic_db.bak	Backup of **lic_db**. Used for recovery. (Encrypted binary file)
cur_db	Run−time cache of current license status. (Encrypted binary file)
log_file	Log of license server events. (Binary file)
user_file	List of users who may and may not obtain licenses. (ASCII file)

Related Information

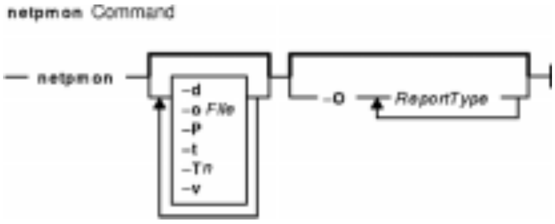
The **glbd** daemon, **llbd** daemon.

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 ...*]

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 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 AIX 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
> ```
>
> will be displayed on standard error, indicating how many trace events were lost while the trace buffers were full. The **netpmon** command will continue monitoring network activity, but the accuracy of the report will be diminished 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.
>
> - 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

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

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

−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 will cause 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 *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.

−t
: Prints CPU reports on a per−thread basis.

−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

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.

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 reveived 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):

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, enter:
   ```
   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, enter:
   ```
   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, enter:
   ```
   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.

Related Information

The **trcstop** command.

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

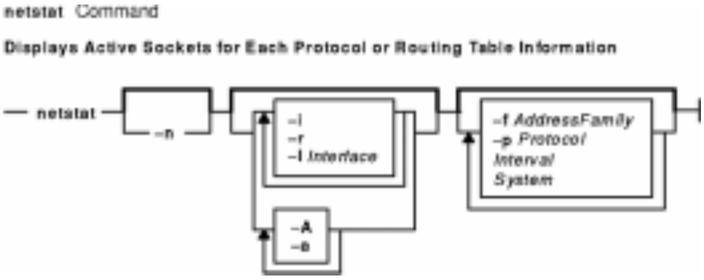
netstat Command

Purpose

Shows network status.

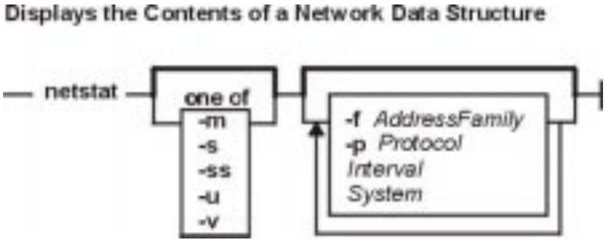
Syntax

To Display Active Sockets for Each Protocol or Routing Table Information



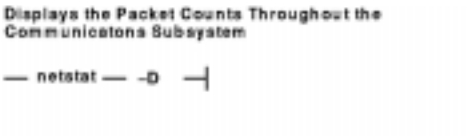
/bin/netstat [**−n**] [{ **−A −a** } |
{ **−r −i −I** *Interface* }] [**−f** *AddressFamily*] [**−p** *Protocol*] [*Interval*] [*System*]

To Display the Contents of a Network Data Structure



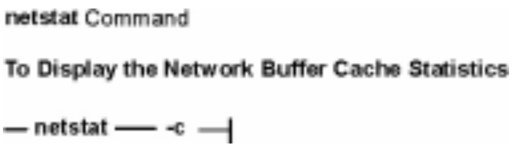
/bin/netstat [**−m** | **−s** | **−ss** | **−u** | **−v**] [
−f *AddressFamily*] [**−p** *Protocol*] [*Interval*] [*System*]

To Display the Packet Counts Throughout the Communications Subsytem



/bin/netstat −D

To Display the Network Buffer Cache Statistics



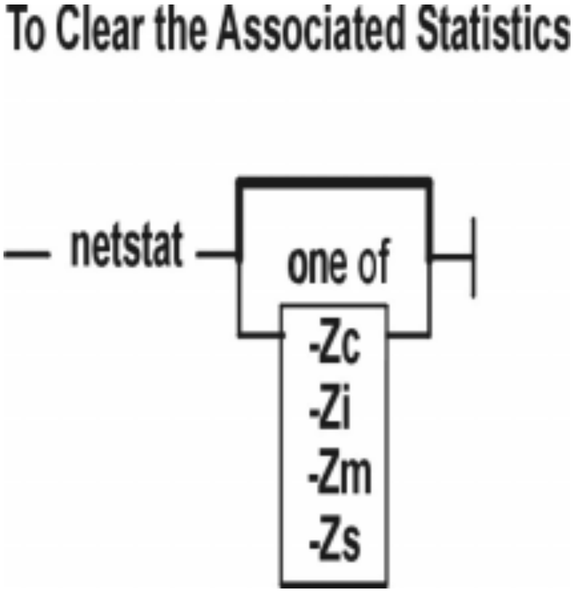
/bin/netstat −c

To Display the Data Link Provider Interface Statistics

netstat Command
To Display the Data Link Provider Interface Statistics
netstat
-P

/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 which contains 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 Version 4.3.2 and above.

−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 which 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.

> **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 Version 4.3.2 and above.

−r Shows the routing tables. When used with the **−s** flag, the **−r** flag shows routing statistics.

−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, enter `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 format indicates the available routes and their statuses. Each route consists of a destination host or network and a gateway to use in forwarding packets.

The routing table contains the following ten fields:

`Flags`

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

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 Version 4.2.1 or later.
W	The route is a cloned route. This field only applies to AIX Version 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.
l	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.

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 Version 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 Version 4.2.1 or later.

`Groups` Provides a list of goup IDs associated with that route. This field only applies to AIX Version 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, enter:

```
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, enter:

   ```
   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, enter:

   ```
   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, enter:

   ```
   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, enter:

   ```
   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 which 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 which are enabled on behalf of IP Multicast for the particular interface.

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

```
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, enter:

   ```
   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, enter:

   ```
   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 Versions 3.2 and 4 Performance Tuning Guide*.

Gateways, Naming, TCP/IP Addressing, TCP/IP Network Interfaces, TCP/IP Protocols, and TCP/IP Routing in *AIX Version 4.3 System Management Guide: Communications and Networks*.

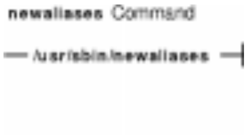
Xerox Network Systems (XNS) Overview for Programming in *AIX Communications Programming Concepts*.

newaliases Command

Purpose

Builds a new copy of the alias database from the **/etc/aliases** file.

Syntax



/usr/sbin/newaliases

Description

The **newaliases** command builds a new copy of the alias database from the **/etc/aliases** file. Running this command is equivalent to running the **sendmail** command with the **−bi** flag.

Files

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

Related Information

The **sendmail** command.

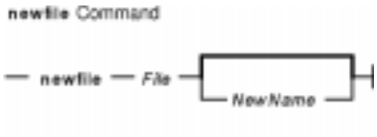
Managing Mail Aliases and How to Build the Alias Database in *AIX Version 4.3 System Management Guide: Communications and Networks*.

newfile Command

Purpose

Converts a text file into an INed structured file.

Syntax



newfile*File* [*NewName*]

Description

The **newfile** command converts a text file into a structured file. If you specify only the file name in the *File* parameter, that will also be the name of the structured file. The original text file is backed up by appending a **.bak** to its name, making it the *File***.bak** file. The **newfile** command exits with an error if the text file does not exist or if it cannot create the structured file.

See the **readfile** command to convert a structured file to a text file.

Examples

1. To convert the `menu2` text file into the `newmenu` structured file, enter:

   ```
   newfile menu2 newmenu
   ```

2. To convert the `menu2` text file into a structured file and save the text version as the `menu2.bak` file, enter:

   ```
   newfile menu2
   ```

Related Information

The **e** command, **ghost** command, **history** command, **readfile** command, **rmhist** command, **versions** command.

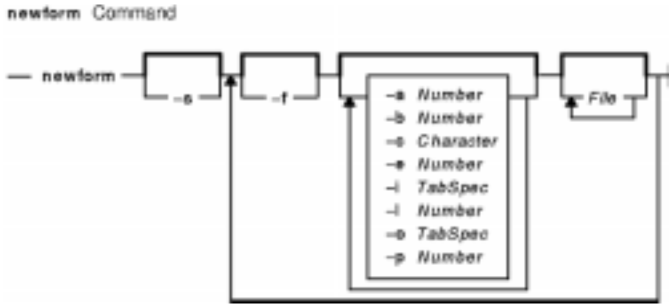
INed Editor Overview in *AIX Version 4.3 INed Editor User's Guide* introduces general concepts about the INed editor.

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. Note, however, that 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

enter 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

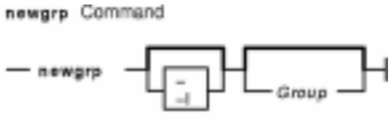
The **tabs** command, **csplit** command.

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

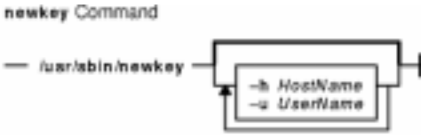
The **login** command, **setgroups** command.

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 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 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.

Setting up and running Web−based System Manager in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

System Management Interface Tool (SMIT): Overview in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

Network File System (NFS) Overview for System Management in *AIX Version 4.3 System Management Concepts: Operating System and Devices*.

How to Export a File System Using Secure NFS, How to Mount a File System Using Secure NFS in *AIX Version 4.3 System Management Concepts: Operating System and Devices*.

Network Information Service (NIS) in *AIX Version 4.3 Network Information Services (NIS and NIS+) Guide*.

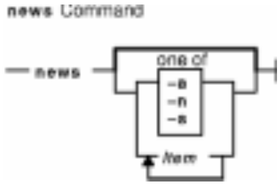
NIS Reference.

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 **msgs** command, **pg** command.

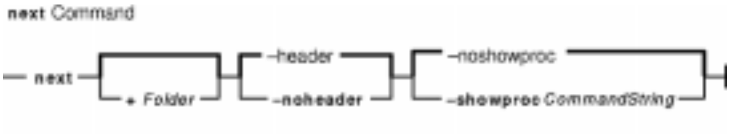
The **/etc/security/environ** file, **profile** file.

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 Version 4.3 System User's Guide: Communications and Networks*.

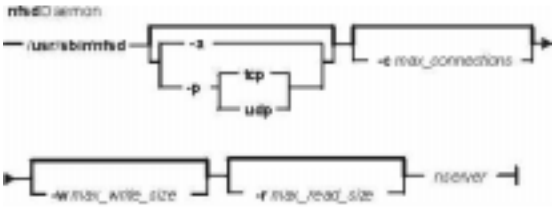
Peek, Jerry. *MH and xmh: E−mail for Users and Programmers.* Sebastopol, CA: O'Reilly & Associates, 1992.

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 Version 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 Version 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 Version 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 Version 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 Version 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 Version 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 Version 4.2.1 or later.

Examples

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

   ```
   startsrc −s nfsd
   ```

   In this example, the `startsrc−snfsd` 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 Version 4.3 System Management Concepts: Operating System and Devices*.

System Resource Controller Overview in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

List of NFS Commands.

nfso Command

Purpose

Configures Network File System (NFS) network variables.

Syntax

nfso Command
—nfso
−a
−d Option
−l HostName
−o Option
= NewValue
−c

nfso { **−a**|**−d***Option* | **−l***HostName* | **−o** *Option* [**=***NewValue*] } [**−c**]

Description

> **Attention:** Be careful when you use this command. The **nfso** command performs no range checking; therefore, it accepts all values for the variables. If used incorrectly, the **nfso** command can make your system inoperable.

Flags

−a
Prints a list of all configurable options and their current values.

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

−d *Option*
Sets the *Option* variable back to its default value.

−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 *Option*[**=***NewValue*]
Shows the value of the option specified by the *Option* parameter if the *NewValue* variable is not specified. If a new value is specified, the *Option* parameter is set to that value.

The **nfso** command sets or displays network options in the kernel. This command operates only on the kernel currently running. Use the **nfso** command after each system startup or network configuration. The effect of changing any value will be immediate unless noted in the description of the option.

These values are valid for the *Options* parameter:

nfs_allow_all_signals
Specifies that the NFS server adhere to signal handling requirements for blocked locks for the UNIX 95/98 test suites. A value of 1 turns **nfs_allow_all_signals** on, and a value of 0 turns it off. The default is off (0).

nfs_device_specific_bufs
Forces the NFS server to use the device−specific buffers. ATM and the SP high speed switch use special buffers for sending data out of the device. The more efficiently the NFS server uses the memory allocations, the better it performs. If resource allocation of the buffers becomes difficult, turn off the **nfs_device_specific_bufs** to diagnose the problem. A value of 1 turns it on, and a value of 0 turns it off. The default is to use the device−specific buffers (1).

nfs_dynamic_retrans
Controls dynamic retransmit packet resizing. The **nfs_dynamic_retrans** option allows the NFS read/write packets to respond to network or server load problems. This option also allows the NFS client to vary the timeouts used for retransmissions, based on the response time of the network of server. The default value is 1 (enabled).

nfs_gather_threshold
Determines when the NFS server attempts to gather write requests to a file. If the size of the NFS write request is less than the value of the **nfs_gather_threshold** option, the NFS server writes the data and immediately responds to the NFS client. If the size of the NFS write request is equal to or greater than the value of this option's value, the NFS server writes the data and waits for a small amount of time before responding to the NFS client. The write−gathering is a performance advantage for sequential writes, but can also be a slight performance disadvantage for random writes. To

turn off write gather, set the **nfs_gather_threshold** to any value greater than 8192. **nfs_gather_threshold** no longer applies to AIX Version 4.2.1 and later versions, but it remains as a compatability interface.

nfs_iopace_pages

Sets the maximum number of dirty pages that the NFS client flushes to the NFS server at one time. The default value is 0, which corresponds to 32 pages. The AIX kernel will modify the default value, depending on write size. However, if you specify a value for **nfs_iopace_pages**, the kernel does not modify that maximum. Valid values are between 1 and 65,536, inclusively.

The **nfs_iopace_pages** option is useful when the binder flushes large compilation images, thus causing interactive performance to suffer. Adjust the value to achieve a balance between binder−write performance and interactive work.

nfs_max_connections

Specifies the maximum number of TCP connections allowed on the NFS server. The default value is 0, which specifies that there is no maximum. One TCP connection is used per client. If new TCP connections are requested from NFS clients and the new connection increases the total amount of connections beyond the maximum, the existing TCP connection closes. The **nfs_max_connections** only applies to AIX Version 4.2.1 or later.

nfs_max_read_size

Sets the maximum and preferred read size. The default size is 32,678 bytes. The maximum is 65,536 bytes and the minimum is 512 bytes. NFS clients mounting after **nfs_max_read_size** is set must use its set value. You cannot use this option to change the size for existing mounts.

Using **nfs_max_read_size** may require you to reduce the V3 read/write sizes when the mounts cannot be manipulated directly in the clients, especially during NIM installations on networks dropping packets with default read/write sizes of 32K. In this case, set the maximum size to a small value.

nfs_max_threads
Specifies the maximum number of **nfsd** threads allowed on an NFS server. In AIX Version 4.3.2 and later versions, the default maximum is 3891. In some systems with migration installs, the system administrator may need to increase the maximum. The maximum number may also be specified for the **nfsd** daemon. The specified value overrides the default. Re−running the **nfso** command to set **nfs_max_threads** overrides the previously set value. This option only applies to AIX Version 4.2.1 or later.

In AIX Version 4.2.1, the NFS server is multi−threaded. Threads are created as demand increases on the server. When the threads are idle, they exit, thus allowing the server to adapt to the needs of the NFS clients. Assuming that NFS serving is the system's primary purpose, setting a large maximum does not detract from overall system performance because the NFS server creates threads as needed. However, if you want to use the system for other activities, you should specify a low value for **nfs_max_threads**.

nfs_max_write_size
Sets the maximum and preferred write size. The default size is 32,678 bytes. The maximum is 65,536 bytes and the minimum is 512 bytes. NFS clients mounting after **nfs_max_write_size** is set must use the set value. You cannot use this option to change the size for existing mounts.

Using **nfs_max_write_size** may require you to reduce the V3 read/write sizes when the mounts cannot be manipulated directly in the clients, especially during NIM installations on networks dropping packets with default read/write sizes of 32K. In this case, set the maximum size to a small value.

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

duplicate NFS messages. When set to a value of 1, all NFS messages are printed to the screen. If set to a value of 0, duplicate messages appearing one after the other are not printed to the screen. Only the first message of such a sequence is displayed. When a different message appears, a message will be displayed similar to:

```
Last NFS message repeated n times.
```

nfs_rfc1323
Enables the use of RFC1323 for NFS sockets. RFC1323 are TCP extensions for use in large bandwidth delay networks as described in the rfc1323. Use this option to enable very large TCP window size negotiations between systems. If using the TCP transport between NFS client and server, **nfs_rfc1323** allows the systems to negotiate a TCP window size so that more data is allowed to be transferred between client and server. In this way, **nfs_rfc1323** increases the throughput potential between client and server. To use this option, both client and server must have TCP support, MTU discovery, and socket buffers large enough to handle the new MTU size.

nfs_server_base_priority
Specifies the base priority for **nfsd** processes. The default value is 0, giving **nfsd** processes regular floating priority. Valid values are between 31 and 126, inclusively. The purpose of **nfs_server_base_priority** is to allow performance tuning of the NFS server or to allow the system administrator to specify a reasonable value, depending on system load requirements.

nfs_server_clread
Allows the NFS server to be very aggressive about reading a file. The NFS server can only respond to the specific read request from the NFS client. However, the NFS server can read existing data in the file immediately after responding to the current read request. This action is commonly called read−ahead. The NFS server performs read−ahead by default. Enabling the **nfs_server_clread** option causes the

NFS server to be very aggressive when doing read−aheads for the NFS client. The default value for this option is 1 (enabled). A value of 0 sets the normal system default for the read−ahead methods.

nfs_setattr_error
Tells the NFS server to ignore invalid *setattr* requests when its value is set to 1. The **nfs_setattr_error** is intended for certain personal computer applications. The default value is 0. This option does not apply to AIX Version 4.2.1 and later versions.

nfs_socketsize
Sets the queue size of the NFS server UDP (User Datagram Protocol) socket. The queue size is specified in number of bytes. The UDP 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_socketsize** option must be less than the **sb_max** option, which can be manipulated by the **no** command.

In AIX Version 4, the socket size is changed dramatically. In this version and later versions, you no longer need to stop and restart the **nfsd** daemon to implement changes to the socket's queue size. The default size is 60,000 bytes. Do not set the **nfs_socketsize** value to less than 60,000. Large and/or busy servers should have values larger than 60,000 until UDP NFS traffic shows no packets dropped in **netstat −s −p udp**.

nfs_tcp_duplicate_cache_size
Specifies the number of entries to store in the NFS server's duplicate cache for the TCP network transport. This option only applies to AIX Version 4.2.1 or later.

nfs_tcp_socketsize
Sets the queue size of the NFS server TCP (Transmission Control Protocol) 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.

In AIX Version 4, the socket size is changed dramatically. In this version and later versions, you no longer need to stop and restart the **nfsd** daemon to implement changes to the socket's queue size. The default size is 60,000 bytes. Do not set the **nfs_tcp_socketsize** value to less than 60,000. Large and/or busy servers should have larger values until TCP NFS traffic shows no packets dropped in **netstat −s −p tcp**.

nfs_udp_duplicate_cache_size
Specifies the number of entries to store in the NFS server's duplicate cache for the UDP network transport. This option only applies to AIX Version 4.2.1 or later.

nfs_use_reserve_ports
Forces the client to use reserved ports for all communication. The default is not to force that use. A value of 1 turns the **nfs_use_reserve_ports** option on, while a value of 0 turns it off.

portcheck
Checks whether an NFS request originated from a privileged port. The default value of 0 disables port checking by the NFS server. A value of 1 directs the NFS server to do port checking on the incoming NFS requests.

udpchecksum
Performs the checksum of NFS UDP packets. The default value of 1 directs the NFS server or client to build UDP checksums for the packets that it sends to the NFS clients or servers. A value of 0 disables the checksum on UDP packets from the NFS server or client. Use **udpchecksum** to check data integrity.

Examples

1. To set the **portcheck** kernel extension variable to a value of zero, enter:

   ```
   nfso −o portcheck=0
   ```

2. To set the **udpchecksum** variable to its default value of 1, enter:

   ```
   nfso −d udpchecksum
   ```

3. To print, in colon−delimited format, a list of all configurable options and their current values, enter:

   ```
   nfso −a −c
   ```

Related Information

The **netstat** command, **no** command.

Network File System (NFS) Overview for System Management in *AIX Version 4.3 System Management Guide: Communications and Networks*.

TCP/IP Overview for System Management in *AIX Version 4.3 System Management Guide: Communications and Networks*.

NFS Tuning in *AIX Versions 3.2 and 4 Performance Tuning Guide*.

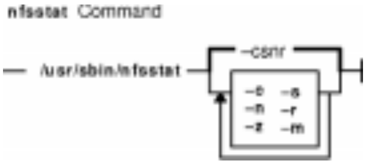
List of NFS Commands.

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 Version 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 Version 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 Version 4.3 System Management Guide: Communications and Networks*.

List of NFS Commands in *AIX Version 4.3 System Management Guide: Communications and Networks*.

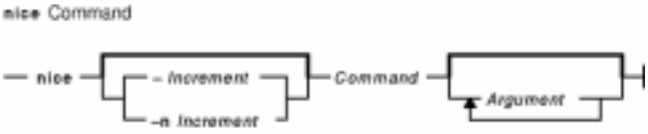
NFS Tuning in *AIX Versions 3.2 and 4 Performance Tuning Guide*.

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 Version 4.3 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 Version 4.3 System User's Guide: Operating System and Devices* discusses what processes are and input and output redirection.

Shells Overview in *AIX Version 4.3 System User's Guide: Operating System and Devices* describes what shells are, the different types, and how they affect the way commands are interpreted.

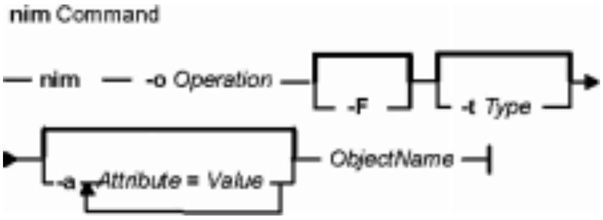
Controlling Contention for the CPU in *AIX Versions 3.2 and 4 Performance Tuning Guide*.

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.

remove
: Removes an object.

reset
: Resets an object's NIM state.

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

unconfig
: Unconfigures the NIM master fileset.

AIX Version 4.2 or later Examples:

maint_boot
: Enables a machine to boot in maintenance mode.

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.

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

AIX Version 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.

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
AIX **mksysb** image.

paging
Parent directory for client paging files.

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

script
Executable file which 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.

AIX Version 4.2 or later Resource Types:

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

resolv_conf
Name−server configuration file. This resource applies only to AIX Version 4.2 or later.

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.

network types:

tok
Token−Ring network.

ent
Ethernet network.

fddi
FDDI network.

atm
ATM network. (AIX Version 4.3 or later.)

generic
Other TCP/IP networks.

AIX Version 4.2 or later only:

group types:

mac_group

Group of machines. This type applies only to AIX Version 4.2 or later.

res_group
Group of resources. This type applies only to AIX Version 4.2 or later.

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 on the token−ring network called `net1` and call it `altoid`, enter:

   ```
   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`, enter:

   ```
   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`, enter:

   ```
   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`, enter:

   ```
   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`, enter:

   ```
   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 which 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` , enter:

   ```
   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, enter:

   ```
   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 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`, enter:

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

remove

To remove a resource named `dump_files`, enter:

```
nim −o remove dump_files
```

change

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`, enter:

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

check

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

   ```
   nim −o check myspot
   ```

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

   ```
   nim −o check images
   ```

allocate

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

   ```
   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 entering:

   ```
   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, enter:

```
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`, enter:

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

   Then enter:

   ```
   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, enter:

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

   Then enter:

   ```
   nim −o cust stand1
   ```

deallocate

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

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

bos_inst

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

```
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 entering:

```
nim −o bos_inst blowfish
```

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`, enter:

   ```
   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, enter:

   ```
   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, enter:

```
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, enter:

   ```
   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, enter:

   ```
   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` , enter:

   ```
   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 entering:

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

   Then, perform the cust operation by entering:

   ```
   nim −o cust catfish
   ```

maint

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

   ```
   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, enter:

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

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

   ```
   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`, enter:

   ```
   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`, enter:

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

dkls_init

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 entering:

```
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 entering:

```
nim −o dkls_init syzygy
```

dtls_init

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 entering:

```
nim −o allocate −a spot=spot1 −a root=root1 −a dump=dump1 syzygy
```

Then initialize the resources for the client machine by entering:

```
nim −o dtls_init syzygy
```

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 enter:

```
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

To check fileset version and requisite consistency on the SPOT `spot1`, enter:

```
nim −o lppchk spot1
```

AIX Version 4.2 or later Examples

allocate

1. 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 enter:

   ```
   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 entering:

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

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

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

bos_inst

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

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

2. To use the default resources when installing the machine `mac1`, enter:

   ```
   nim −o bos_inst mac1
   ```

change

1. 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`, enter:

   ```
   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.

2. To define default routes for the networks `net1` and `net2` which use default gateways `gw1` and `gw2` respectively, enter the following two commands:

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

3. 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, enter:

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

cust

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

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

2. 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, enter:

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

define

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

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

2. To define a resource group named `DisklsRes1` with resources `spot1`, `root1`, `dump1`, `paging1`, `home1`, `tmp1`, enter:

   ```
   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
   ```

3. 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, enter:

   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
   ```

4. 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, enter:

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

5. 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, enter:

   ```
   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
   ```

6. To define an **exclude_files** resource, `exclude_file1`, located in `/resources/exclude_file1` on the master, enter:

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

7. 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, enter:

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

8. 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`, enter:

   ```
   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.

dkls_init

1. To initialize the environment for a diskless workstation with the name of `syzygy`, enter:

   ```
   nim −o dkls_init syzygy
   ```

2. 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`, enter the following two commands:

   ```
   nim −o select −a exclude=diskls2 DisklsMacs1
   nim −o dkls_init −a group=DisklsRes1 DisklsMacs1
   ```

3. 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, enter:

   ```
   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`, enter:

   ```
   nim −o dtls_init syzygy
   ```

2. 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`, enter the following two commands:

   ```
   nim −o select −a exclude=datals2 DatalsMacs1
   nim −o dtls_init −a group=DatalsMacs1 DatalsRes1
   ```

3. 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, enter:

   ```
   nim −o dtls_init −a group=DatalsMacs1 DatalsRes1
   ```

lppchk

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, enter:

```
nim −o lppchk −a lppchk_flags='−c −m3 −u' \
−a filesets='bos*' macgrp1
```

Since the **lppchk** operation runs in the background on group members by default, to view the output from the **lppchk** operation enter:

```
nim −o showlog −a log_type=lppchk macgrp1
```

maint

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, enter:

```
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, enter:

```
nim −o maint_boot stand1
```

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

showlog

To view the boot logs of the machines defined by the group `DisklsMacs1`, enter:

```
nim −o showlog −a log_type=boot DisklsMacs1
```

showres

1. To show the contents of the config script `script1`, enter:

   ```
   nim −o showres script1
   ```

2. To show the contents of the bosinst.data resource `bosinst_data1`, enter:

   ```
   nim −o showres bosinst_data1
   ```

3. To list all the filesets in the lpp_source `lpp_source1`, enter:

   ```
   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`, enter:

   ```
   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`, enter:

   ```
   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 Version 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, enter:

   ```
   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), enter:

   ```
   nim −o alt_disk_install −a source=rootvg\
   −a disk='hdisk2'\
   −a phase=12 austin
   ```

Files

/etc/niminfo Contains variables used by NIM.

Related Information

The **lsnim** command, **nimclient** command, **nimconfig** command, **niminit** command.

The **.info** file.

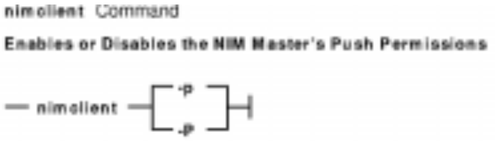
nimclient Command

Purpose

Allows Network Installation Management (NIM) operations to be performed from a NIM client.

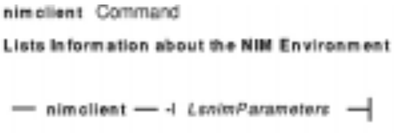
Syntax

To Enable or Disable the NIM Master's Push Permissions



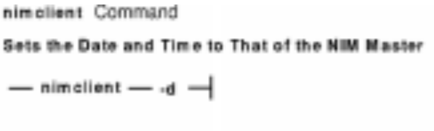
nimclient { **−p** } | { **−P** }

To List Information about the NIM Environment



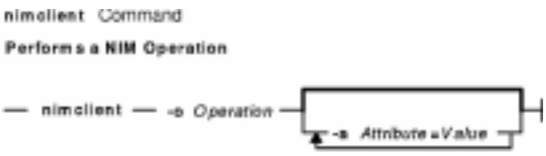
nimclient−l*LsnimParameters*

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.

AIX Version 4.2 or later Operations:

maint_boot	Enables a machine to boot in maintenance mode. This operation only applies to AIX Version 4.2 or later.
reset	Resets an object's NIM state.
showres	Displays the contents of a NIM resource. This operation applies only to AIX Version 4.2 or later.

−p Enables the NIM master to push commands.

−P Removes the NIM master's permissions to push commands.

Note: The master can overrride 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
    ```

AIX Version 4.2 or later Examples

1. 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.

2. To show the contents of the config script `script1`, enter:

   ```
   nimclient −o showres −a resource=script1
   ```

3. To show the contents of the bosinst.data resource `bosinst_data1`, enter:

   ```
   nimclient −o showres −a resource=bosinst_data1
   ```

4. 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.

5. 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"
   ```

6. 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
   ```

7. 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
   ```

8. 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
   ```

9. 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.

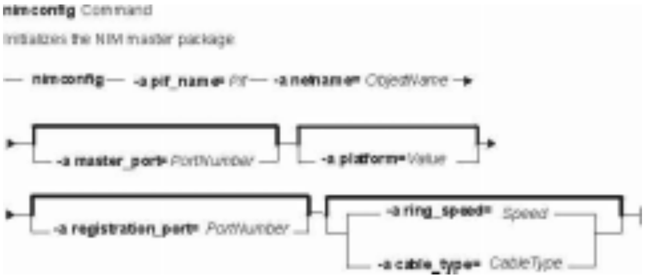
nimconfig Command

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 which 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) which 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

rs6ksmp
Micro Channeled−based, symmetric multiprocessor models

rspc
PowerPC PCI bus−based, uniprocessor models

rspcsmp
PowerPC PCI bus−based, symmetric multiprocessor models

netname=*ObjectName*
Specifies the name you want the **nimconfig** command to use when creating the network object to represent the network which the master's primary interface connects to.

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

AIX Version 4.2 or later Attributes:

registration_port=*PortNumber*
Specifies the port number used for NIM client registration. This attribute applies only to AIX Version 4.2 or later.

> **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 which 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, enter:

   ```
   nimconfig −a pif_name=tr0 −a netname=net1 −a ring_speed=16
   ```

2. To initialize the NIM environment using ethernet and the default NIM ports, enter:

   ```
   nimconfig −a pif_name=en0 −a master_port=1058 \
   −a netname = net2 −a cable_type=bnc
   ```

3. To rebuild the **/etc/niminfo** file on the NIM master when that machine has already been correctly configured as a master, enter:

   ```
   nimconfig −r
   ```

4. To initialize the NIM master using an ATM network interface, enter:

   ```
   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.

AIX Version 4.2 or later Examples

1. To initialize the NIM environment using TCP/IP port 1060 for NIM client communications and TCP/IP port 1061 for NIM client registration, enter:

   ```
   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.

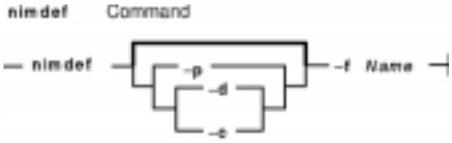
The **.info** file.

nimdef Command

Purpose

Defines Network Installation Management (NIM) clients from a stanza file. This command only applies to AIX Version 4.2 or later.

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 Version 4.3 Network Installation Management 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**.

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.

comment 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 mutiple 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

```
# 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:
    comment = "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.

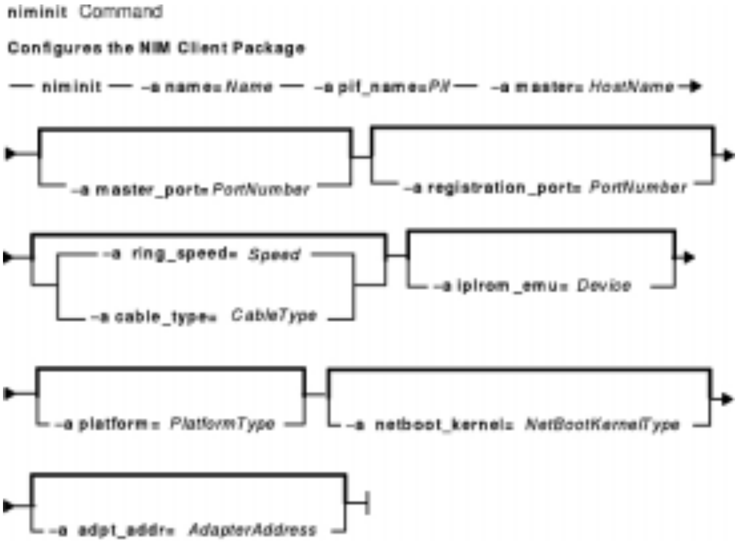
niminit Command

Purpose

Configures the Network Installation Management (NIM) client package.

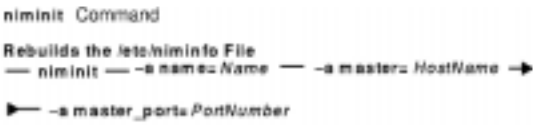
Syntax

To Configure the NIM Client Package


niminit Command
Configures the NIM Client Package
niminit
-a name= Name
-a pif_name=Pif
-a master= HostName
-a master_port= PortNumber
-a registration_port= PortNumber
-a ring_speed= Speed
-a cable_type= CableType
-a iplrom_emu= Device
-a platform= PlatformType
-a netboot_kernel= NetBootKernelType
-a adpt_addr= AdapterAddress

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*] [**−aplatform=***PlatformType*] [**−a netboot_kernel=***NetbootKernelType*] [**−aadpt_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.

Once the NIM client package has been successfully configured, the **niminit** command may 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

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

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
rs6ksmp	Micro Channel−based, symmetric multiprocessor models
rspc	PowerPC PCI bus−based, uniprocessor machines
rspcsmp	PowerPC PCI bus−based, symmetric multiprocessor machines

adpt_add=*AdapterAddress*
Specifies the hardware address that corresponds to the network adapter.

AIX Version 4.2 or later Attributes:

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 which 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, enter:

   ```
   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`, enter:

   ```
   niminit −a name=robin −a master=superman −a master_port=1058
   ```

3. To configure the NIM client package 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`, enter:

   ```
   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`, enter:

   ```
   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 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`, enter:

   ```
   niminit −a name=jellyfish −a pif_name=en0 −a cable_type=dix \
   −a master=whale −a master_port=1058 −a platform=rspcsmp
   ```

AIX Version 4.2 or later Examples:

1. To configure the NIM client package on a machine which 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, enter:

   ```
   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
   ```

File

/etc/niminfo Contains variables used by NIM.

Related Information

The **lsnim** command, **nim** command, **nimclient** command, **nimconfig** command.

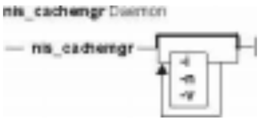
The **.info** file.

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.

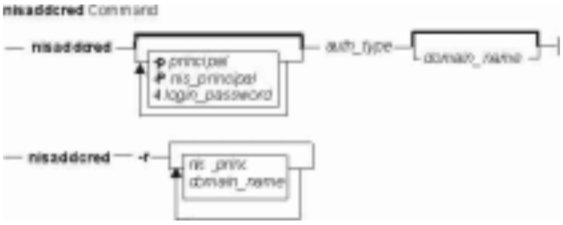
The **rpc.nisd** daemon .

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

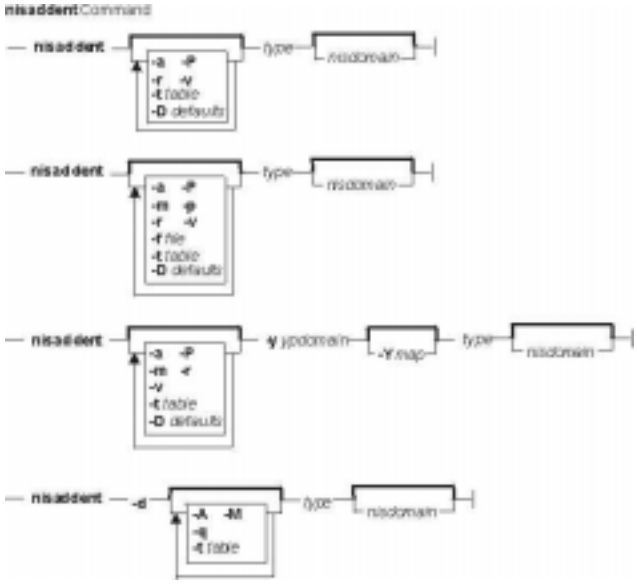
The **chkey** command, **domainname** command, **keylogin** command, **niscat** command, **nischmod** command, **nischown** command, **nisclient** command, **nismatch** command, **nispopulate** command, **ps** command.

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

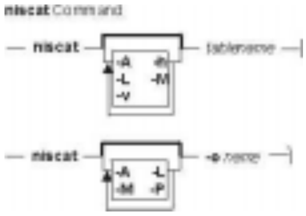
The **niscat** command, **nischmod** command, **nisdefaults** command, **nispopulate** command, **nisserver** command, **nissetup** command, **nistbladm** command, **passwd** command, **ypxfr** command.

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 `#' (hash) 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, enter:
   ```
   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, enter:
   ```
   niscat passwd.org_dir
   ```

3. To display the contents of table **frodo** and the contents of all tables in its concatenation path, enter:
   ```
   niscat −A frodo
   ```

4. To display the entries in the table groups.org_dir as NIS+ objects, enter:
   ```
   niscat −o '[ ]groups.org_dir'
   ```

   The brackets are protected from the shell by single quotes.

5. To display the table object of the passwd.org_dir table, enter:
   ```
   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, enter:
   ```
   niscat −o org_dir
   ```

Related Information

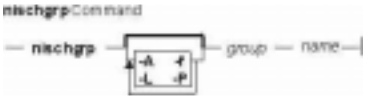
The **nistbladm** command, **nisdefaults** command, **nismatch** command.

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

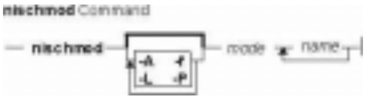
The **nischmod** command, **nischown** command, **nisdefaults** command, **nisgrpadm** command.

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

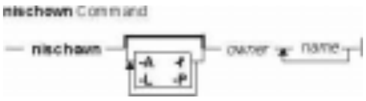
The **chmod** command, **nischgrp** command, **nischown** command, **nisdefaults** command.

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.

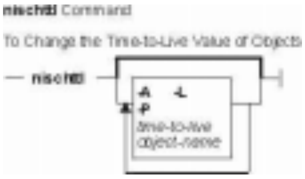
nischttl Command

Purpose

The **nischttl** command changes the *time−to−live* value of objects or entries in the namespace.

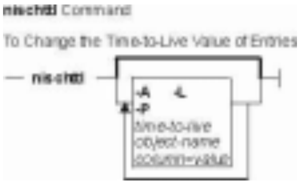
Syntax

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.

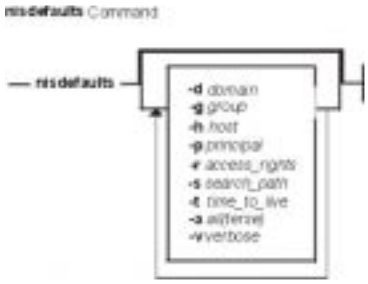
resolv.conf file format.

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. Once 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 below:

```
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

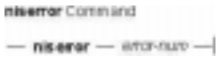
1. To display all values in verbose format, type the **nisdefaults** command without arguments.
```
master% nisdefaults
Principal Name : topadmin.wiz.com.
Domain Name    : Wiz.com.
Host Name      : rootmaster.wiz.com.
Group Name     : salesboss
Access Rights  : −−−−rmcdr−−−r−−−
Time to live   : 12:00:00:00:00
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:

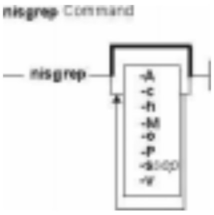
```
niserror 20
Not Found, no such name
```

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.

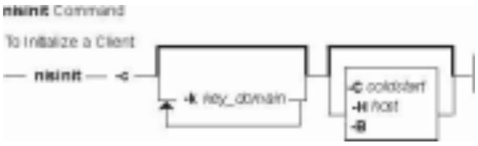
nisinit Command

Purpose

Initializes a workstation to be a NIS+ client.

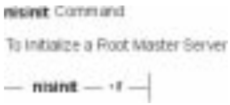
Syntax

To Initialize a Client



nisinit [**−c** [**−k***key_domain*] [**−C***coldstart* | **−H***host*| **−B**]]

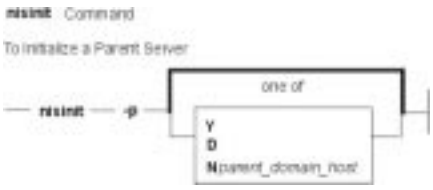
To Initialize a Root Master Server



nisinit−r

To Initialize a Parent Server

[**−pY**|



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

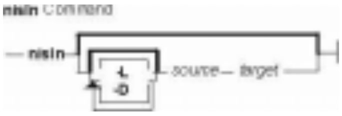
The **nisclient** command, and **nisshowcache** command.

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.

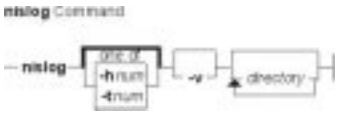
```
nisln source−object target−object
nisln [column=value,...],tablename target−object
```

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

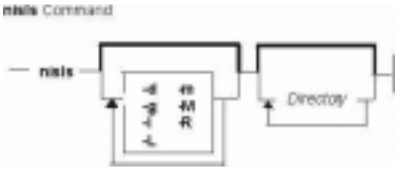
−h *num*	Display transactions starting with the head (beginning) of the log. If the number is omitted, the display begins with the first transaction. If the number 0 is entered, only the log header is displayed
−t*num*	Display transactions starting backward from the end (tail) of the log. If the number 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

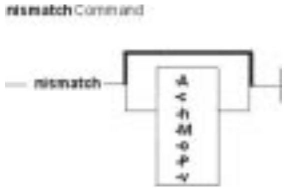
The **nisdefaults** command, **nisgrpadm** command, **nismatch** command, and **nistbladm** command.

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

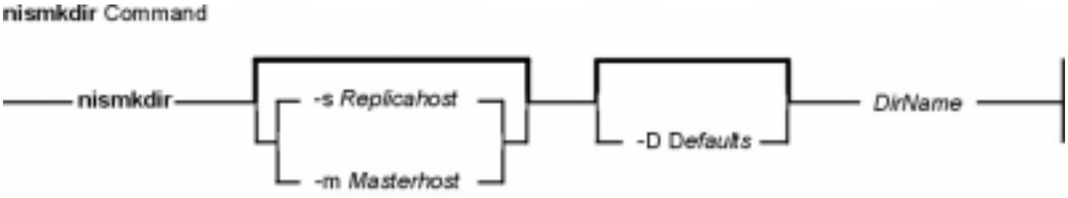
The **nisgrep** command, **nisdefaults** command, **niscat** command, **nisls** command, and **nistbladm** command.

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.

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.

> **Note:** Although this command will allow the user to set the "home" directory for the NIS+ user, no actual physical directory will be created if the directory does not already exist.

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 Security Administration in the *AIX Version 4.3 System Management Guide: Operating System and Devices*.

Setting up and running Web−based System Manager in the *AIX Version 4.3 System Management Guide: Operating System and Devices*.

For more information about administrative roles, refer to Administrative Roles Overview in the *AIX Version 4.3 System Management Guide: Operating System and Devices*.

nisping Command

Purpose

Pings replica servers, telling them to ask the master server for updates immediately.

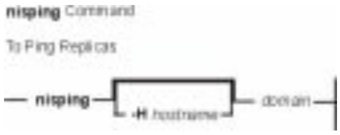
Note: The replicas normally wait a couple of minutes before executing this request.

Syntax

To Display the Time of the Last Update

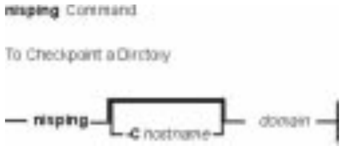
nisping [**−u***domain*]

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:

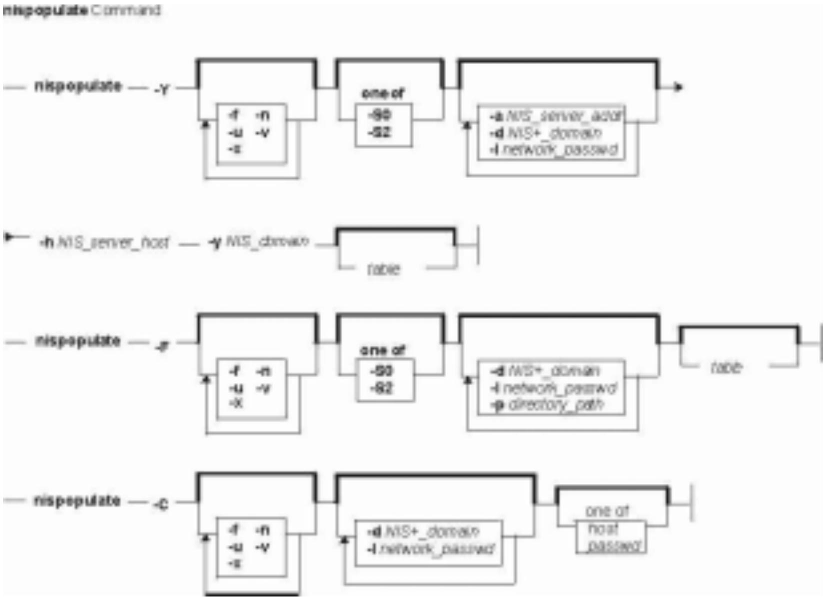
```
rootmaster# /usr/lib/nis/nisping −C
Checkpointing replicas serving directory wiz.com. :
 Master server is rootmaster.wiz.com.
        Last update occurred at Wed May 25 10:53:37 1995
 Master server is rootmaster.wiz.com.
 checkpoint has been scheduled with rootmaster.wiz.com.
 Replica server is rootreplica1.wiz.com.
        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**] [**−S0** | **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**] [**−S0** | **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

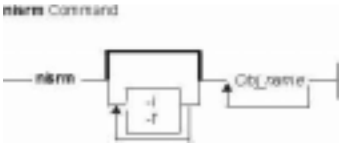
The **nistbladm** command, **nisaddcred** command, **nisaddent** command, **nisclient** command, **nisserver** command, **nissetup** command, **rpc.nisd** command, **ypxfr** command.

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

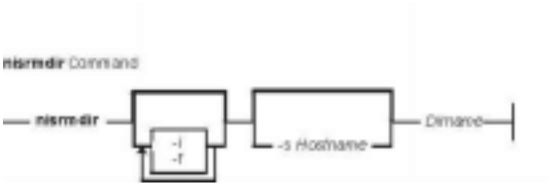
The **nischmod** command, **nisdefaults** command, **nisrmdir** command, **nistbladm** command, and the **rm** command.

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

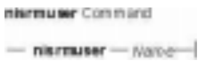
The **nisdefaults** command and the **nisrm** command.

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 (**websm**) 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 Security Administration in the *AIX Version 4.3 System Management Guide: Operating System and Devices*.

For more information about administrative roles, refer to Administrative Roles Overview in the *AIX Version 4.3 System Management Guide: Operating System and Devices*.

Setting up and running Web−based System Management in the *AIX Version 4.3 System Management Guide: Operating System and Devices*.

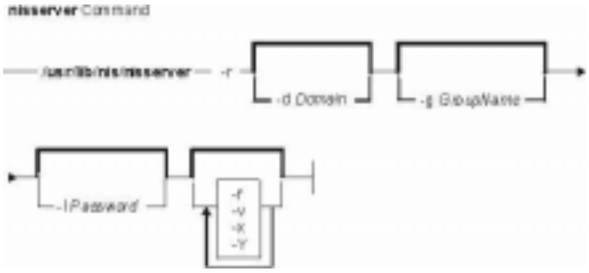
nisserver Command

Purpose

Sets up NIS+ servers.

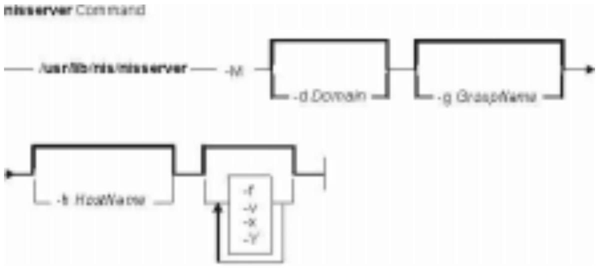
Syntax

To set up a root master server



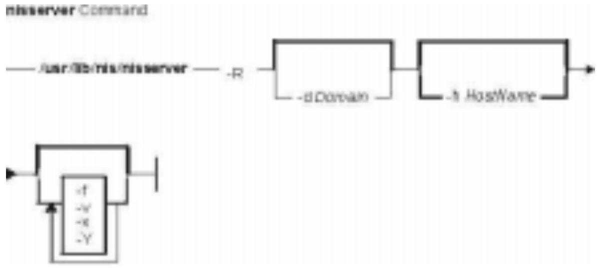
/usr/lib/nis/nisserver−r [**−d***Domain*] [**−f**] [**−g***GroupName*] [**−l***Password*] [**−v**] [**−x**] [**−Y**]

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

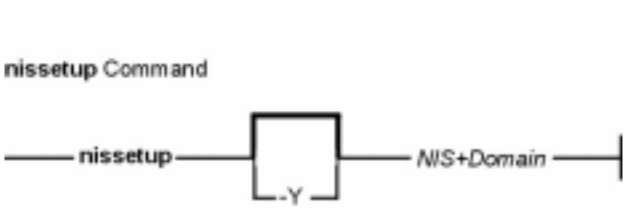
The **nisaddcred** command, the **nisclient** command, the **nisgrpadm** command, the **nisinit** command, the **nismkdir** command, the **nispopulate** command, the **nissetup** command, and the **rpc.nisd** daemon.

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

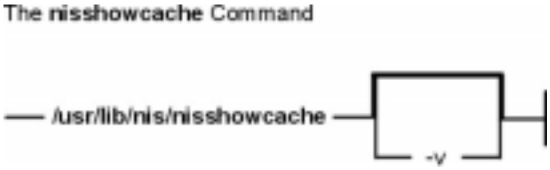
The **nisaddent** command, the **nisinit** command, the **nismkdir** command, and the **nistbladm** command.

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

The **nisstat** Command
/usr/lib/nis/nisstat
-H *HostName*
DirName

/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.

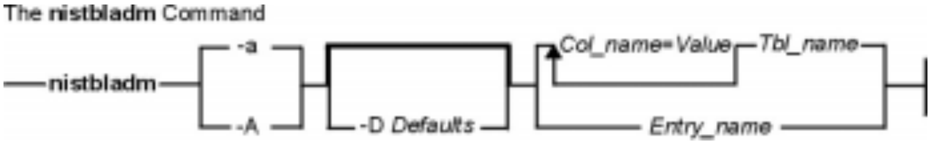
nistbladm Command

Purpose

Administers NIS+ tables.

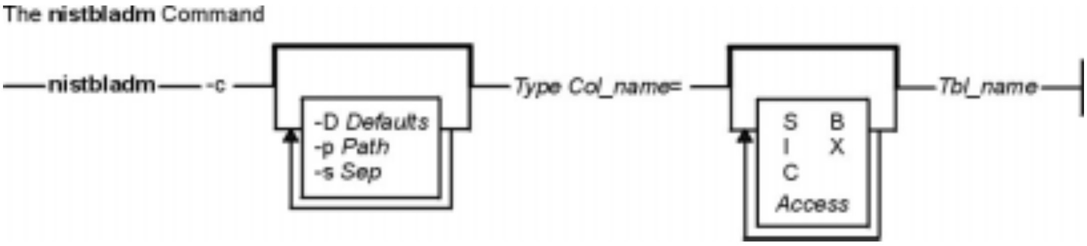
Syntax

To add or overwrite table entries



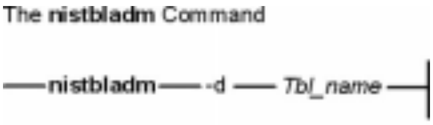
nistbladm−a | **−A** [**−D***Defaults*] { *Col_name=Value*... *Tbl_name* } | { *Entry_name* }

To create an NIS+ table



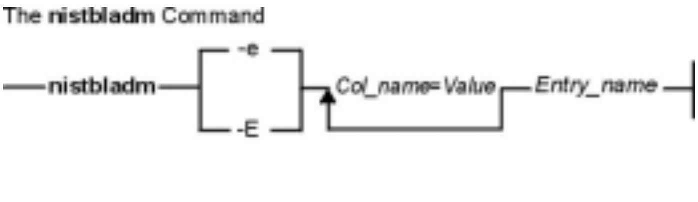
nistbladm−c [**−D***Defaults*] [**−p***Path*] [**−s***Sep*] *Type Col_name*=[**S**] [**I**] [**C**] [**B**] [**X**]
[*Access*]... *Tbl_name*

To delete an entire NIS+ table



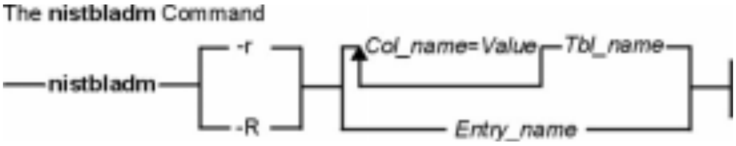
nistbladm−d*Tbl_name*

To edit table entries



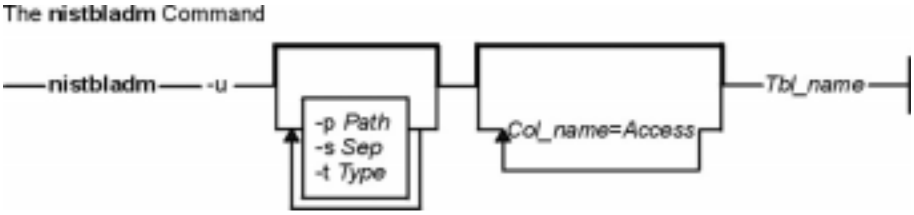
nistbladm−e | **−E***Col_name=Value*... *Entry_name*

To remove table entries



nistbladm−r | **−R** { [*Col_name=Value*...] *Tbl_name* } | { *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, you should 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, you should 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 Addds 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 quotes (') or double quotes ("). 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. **nistbladm** 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 Speciies 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.

−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`, enter:

   ```
   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, enter:

   ```
   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, enter:

   ```
   nistbladm −u −p hobbies.xyz.com.:hobbies.def.com. hobbies
   ```

4. To delete skiing−enthusiasts from the table, enter:

   ```
   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, enter:

   ```
   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

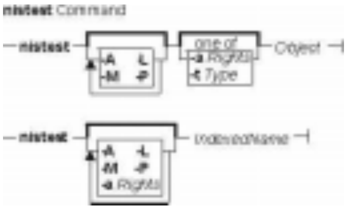
The **niscat** command, the **nischmod** command, the **nischown** command, the **nisdefaults** command, the **nismatch** command, and the **nissetup** command.

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

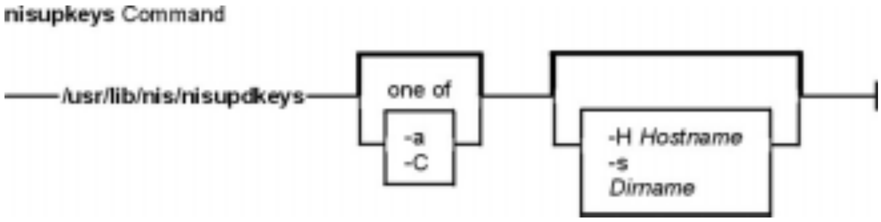
The **nischmod** command and **nisdedaults** command.

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.

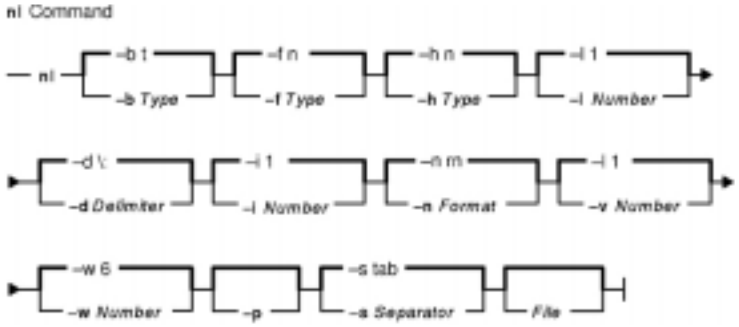
The **gethostbyname** subroutine.

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 test 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 Version 4.3 System User's Guide: Operating System and Devices*.

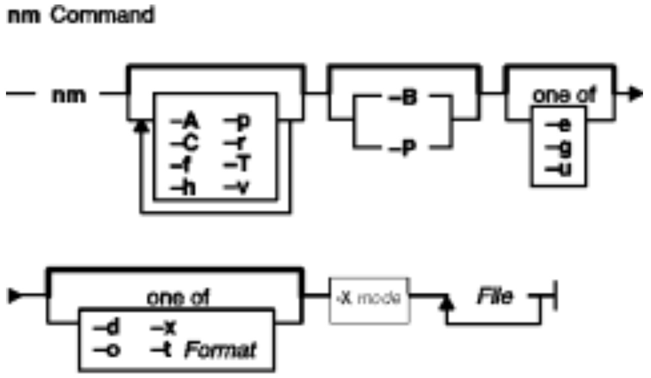
The **pr** command.

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**] [**−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:

  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.

−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 withe 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.

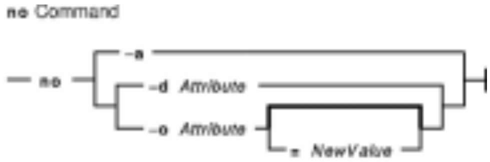
Commands Overview in *AIX Version 4.3 System User's Guide: Operating System and Devices*.

no Command

Purpose

Configures network attributes.

Syntax



no {**−a** | **−d***Attribute* | **−o** *Attribute* [**=***NewValue*] }

Description

Use the **no** command to configure network attributes. The **no** command sets or displays current network attributes in the kernel. This command only operates on the currently running kernel. The command must be run again after each startup or after the network has been configured. Whether the command sets or displays an attribute is determined by the accompanying flag. The **−o** flag performs both actions. It can either display the value of an attribute or set a new value for an attribute. For a more information on how the network attributes interact with each other, refer to the *AIX Version 4.3 System Management Guide: Communications and Networks*.

> **Attention:** Be careful when you use this command. The **no** command performs no range checking, therefore it accepts all values for the variables. If used incorrectly, the **no** command can cause your system to become inoperable.

Some network attributes are runtime attributes that can be changed at any time. Others are loadtime attributes that must be set before the netinet kernel extension is loaded and be placed near the rop of **/etc/rc.net**. If your system uses Berkeley−style network configuration, set the attributes near the top of **/etc/rc.bsdnet**.

Flags

−a	Prints a list of all configurable attributes and their current values.
−d *Attribute*	Sets *Attribute* back to its default value.
−o *Attribute* [**=***NewValue*]	Displays the value of *Attribute* if *NewValue* is not specified else sets *Attribute* to *NewValue*. **Note:** When using the **−o** flag do not enter space characters before or after the equal sign. If you do, the command will fail.

Attributes

You can set the following attributes:

arpqsize Specifies the maximum number of packets to queue while waiting for ARP responses. Default value is 1. This attribute 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 attribute applies to AIX Version 4.1.5, 4.2.1, and later. **arpqsize** is a runtime attribute.

arptab_bsiz
Specifies Address Resolution Protocol (ARP) table bucket size. The default value is 7. **arptab_bsiz** is a loadtime attribute.

arptab_nb
Specifies the number of ARP table buckets. The default value is 25. **arptab_nb** is a loadtime attribute.

arpt_killc
Specifies the time in minutes before a complete ARP entry will be deleted. The default value is 20 minutes. **arpt_killc** is a runtime attribute.

bcastping
Allows response to ICMP echo packets to the broadcast address. A value of 0 turn it off; while a value on 1 turns it on. Default is 0. **bcastping** is a runtime attribute.

clean_partial_conns
Specifies whether or not we are avoiding SYN attacks. If non−zero, **clean_partial_conns** specifies how many partial connections to randomly remove to make room for new non−attack connections. This is a runtime attribute. The default is 0, off.

delayack
Delays ACKs for certain TCP packets and attempts to piggyback them with the next packet sent instead. This will only be performed for connections whose destination port is specified in the list of the **delayackports** attribute. This can be used to increase performance when communicating with an HTTP server. This attribute is available only in AIX 4.3.2 and beyond. The attribute can have one of four values:

0 No delays; normal operation

1 Delay the ACK for the server's SYN

2 Delay the ACK for the server's FIN

3 Delay both the ACKs for the SYN and FIN

delayackports
Specifies the list of destination ports for which the operation defined by the delayack port option will be performed. The attribute 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 **{}** This attribute is available only in AIX 4.3.2 and beyond..

directed_broadcast
Specifies whether or not to allow a directed broadcast to a gateway. 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. **directed_broadcast** is a runtime attribute.

extendednetstats
Enables more extensive statistics for network memory services. The default for this attribute is 1. However, because these extra statistics cause a reduction in system performance,

extendednetstats is set to 0, for off, in **/etc/rc.net**. If these statistics are desired, it is recommended that the code in **/etc/rc.net** that sets **extendednetstats** to 0 be commented out. This attribute is available only in AIX 4.3.2 and beyond.

fasttimo
Allows you to set the millisecond delay for the TCP fast timeout timer. Its range is 50 to 200 milliseconds. Reducing this timer value may improve performance with some non−IBM systems. However, this may also result in slightly increased system utilization.

icmpaddressmask
Specifies whether the system responds to an ICMP address mask request. If the default value 0 is set, the network silently ignores any ICMP address mask request that it receives. **icmpaddressmask** is a runtime attribute.

ie5_old_multicast_mapping
Specifies IP multicasts on token ring should be mapped to the broadcast address rather than a functional address when value 1 is used. The default value is 0. **ie5_old_multicast_mapping** is a runtime attribute.

ifsize
Specifies the maximum number of network interface structures per interface. The default value is 8. In AIX 4.3.2 and above, if the system detects at boot time that more adapters of a type are present than would be allowed by the current value of **ifsize**, it will automatically increase the value to support the number of adapters present. **ifsize** is a loadtime attribute.

inet_stack_size
Lets you configure the inet interrupt stack table size. This is needed if you were running with unoptomized debug kernel and/or netinet. It must be set in **rc.net**; changing it on the fly has no effect. This is different from the pin more stack code because this is on interrupt. The pin more stack code is not configurable. **inet_stack_size** is specified in KB, the default is 16 KB.

ipforwarding
Specifies whether the kernel should forward packets. The default value of 0 prevents forwarding of IP packets when they are not for the local system. A value of 1 enables forwarding. **ipforwarding** is a runtime attribute.

ipfragttl
Specifies the time to live for IP fragments. The default value is 60 half−seconds. **ipfragttl** is a runtime attribute.

ipignoreredirects
Specifies whether or not to process redirects that are received. The default value of 0 processes redirects as usual. A value of 1 ignores redirects. This option only applies to AIX Version 4.2.1 or later. **ipignoreredirects** is a runtime attribute.

ipqmaxlen
Specifies the number of received packets that can be queued on the IP protocol input queue. **ipqmaxlen** is a loadtime attribute.

ipsendredirects
Specifies whether the kernel should send redirect signals. The default value of 1 sends redirects. A value of 0 does not send redirects. **ipsendredirects** is a runtime attribute.

ipsrcrouteforward
Specifies whether the system forwards source routed packets. 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 attribute only applies to AIX Version 4.2.1 or later.

ipsrcrouterecv
Specifies whether the system accepts source routed packets.

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 attribute only applies to AIX Version 4.2.1 or later.

ipsrcroutesend
Specifies whether applications can send source routed packets. 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 attribute only applies to AIX Version 4.2.1 or later.

ip6_defttl
Specifies the default hop count that is used for IPv6 packets if no other hop count is specified.

ip6forwarding
Specifies whether the kernel should forward ipv6 packets. The default value of 0 prevents forwarding of ipv6 packets when they are not for the local systems. A value of 1 enables forwarding. This is a runtime attribute.

ip6_prune
Specifies how often to check the IPv6 routing table for expired routes. The default is 2 seconds.

ip6srcrouteforward
Specifies whether the system forwards source−routed IPv6 packets. 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.

main_if6
Specifies the interface to use for link local addresses. This is only used by **autoconf6** to setup initial routes.

main_site6
Specifies the interface to use for site local address routing. This is only used if **multi_homed** is set to 3.

maxnip6q
Specifies the maximum number of ipv6 packet reassembly queues. The default is 20.

maxttl
Specifies the time to live for RIP packets. The default is 255 seconds. **ipsendredirects** is a runtime attribute.

multi_homed
Specifies the level of multi−homed ipv6 host support.

0 Indicates the original functionality in AIX 4.3.0.

1 Indicates that link local addresses will be a 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
Specifies the total maximum amount of memory that can be used for the Network Buffer Cache. This attribute is in number of KBytes. The default value is derived from **thewall**. When the cache grows to this limit, the least−used cache objects are flushed out of cache to make room for the new ones. This attribute only applies to AIX Version 4.3.2 or later.

nbc_max_cache
Specifies the maximum size of the cache object allowed in the Network Buffer Cache without using the private segments. This

parameter is in number of bytes, the default being 131,072 (128K) bytes. A data object bigger than this size is either cached in a private segment or is not cached at all. This attribute only applies to AIX Version 4.3.2 or later.

nbc_min_cache
Specifies the minimum size of the cache object allowed in the Network Buffer Cache. This attribute is in number of bytes, the default being 1 byte. A data object smaller than this size is not put into the NBC. This attribute only applies to AIX Version 4.3.2 or later.

nbc_pseg
Specifies the maximum number of private segments that can be created for the Network Buffer Cache. The default value is 0. When this option is set at non−0, a data object between the size specified in **nbc_max_cache** and the segment size (256MB) is cached in a private segment. A data object bigger than the segment size is not cached at all. When the maximum number of private segments exist, cache data in private segments may be flushed for new cache data so that the number of private segments do not exceed the limit. When **nbc_pseg** is set to 0, all cache in private segments are flushed. This attribute only applies to AIX Version 4.3.3 or later.

nbc_pseg_limit
Specifies the maximum amount of cached data size allowed in private segments in the Network Buffer Cache. This value is expressed in KBytes. The default value is half of the total real memory size on the running system. Since data cached in private segments are pinned by the Network Buffer Cache, **nbc_pseg_limit** controls the amount of pinned memory used for the Network Buffer Cache in addition to the network buffers in global segments. When the amount of cached data reaches this limit, cache data in private segments may be flushed for new cache data so that the total pinned memory size doesn't exceed the limit. When **nbc_pseg_limit** is set to 0, all cache in private segments is flushed. This attribute only applies to AIX Version 4.3.3 or later.

ndpqsize
Specifies the number of packets to hold waiting on completion of a Neighbor Discovery Protocol (NDP) entry. The default is 50 packets.

ndpt_down
Specifies the time, in half seconds, to hold down a NDP entry. The default value is 3 units, or 1.5 seconds.

ndpt_keep
Specifies the time, in half seconds, to keep a NDP entry. The default value is 120 or 60 seconds.

ndpt_mmaxtries
Specifies the maximum number of Multicast NDP packets to send. The default is value is 3.

ndpt_probe
Specifies the time, in half seconds, to delay before sending their first NDP probe. The default value is 5 units, or a 2.5 seconds.

ndpt_reachable
Specifies the time, in half seconds, to test if a NDP entry is still valid. The default is 30, or 15 seconds.

ndpt_retrans
Specifies the time, in half seconds, to wait before retransmitting a NDP request. The default is 1, or a half second.

ndpt_umaxtries
Specifies the maximum number of Unicast NDP packets to send. The default is value is 3.

net_malloc_police
Specifies the size of the `net_malloc/net_free` trace buffer. If the value of this variable is non−zero all `net_malloc` and `net_free`'s 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. 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. **net_malloc_police** is a runtime attribute.

nonlocsrcroute
Tells the Internet Protocol that strictly source−routed packets may be addressed to hosts outside the local network. 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 attribute. **nonlocsrcroute** is a runtime attribute.

pmtu_default_age
Specifies the default amount of time (in minutes) before the path MTU value for UDP paths is checked for a lower value. 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 attribute only applies to AIX Version 4.2.1 or later. **pmtu_default_age** is a runtime attribute.

pmtu_rediscover_interval
Specifies the default amount of time (in minutes) before the path MTU value for UDP and TCP paths are checked for a higher value. A value of 0 allows no path MTU rediscovery. The default value is 30 minutes. This attribute only applies to AIX Version 4.2.1 or later. **pmtu_rediscover_interval** is a runtime attribute.

rfc1122addrchk
Performs address validation as specified by RFC1122, *Requirements for Internet Hosts−Communication Layers*. The default value of 0 does not perform address validation. A value of 1 performs address validation. **rfc1122addrchk** is a runtime attribute.

rfc1323
Enables TCP enhancements as specified by RFC 1323, *TCP Extensions for High Performance*. The default value of 0 disables the RFC enhancements on a system−wide 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. **rfc1323** is a runtime attribute. In AIX Version 4.3.3 and later versions, the **rfc1323** network option can also be set on a per interface basis via the **ifconfig** command.

route_expire
Specifies whether the route expires. A value of 0 allows no route expiration, which is the default. Negative values are not allowed for this option. This attribute only applies to AIX Version 4.2.1 or later. **route_expire** is a runtime attribute. In AIX Version 4.3.3 and later versions, the default value is 1.

routerevalidate
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) will use the correct route after routing table changes occur. The default value of 0 does not revalidate the cached routes.

Turning this option on may cause some performance degradation. **routerevalidate** is a runtime attribute.

rto_length
Specifies the TCP Retransmit Time Out length value used in calculating factors and the maximum retransmits allowable used in TCP data segment retransmits. `rto_length` is the total number of time segments. Default is 13. **rto_length** is a loadtime attribute.

rto_limit
Specifies the TCP Retransmit Time out limit value used in calculating factors and the maximum retransmits allowable used in TCP data segment retransmits. `rto_limit` is the number of time segments from `rto_low` to `rto_high`. Default is 7. **rto_limit** is a loadtime attribute.

rto_low
Specifies the TCP Retransmit Time Out low value used in calculating factors and the maximum retransmits allowable used in TCP data segment retransmits. `rto_low` is the low factor. Default 1. **rto_low** is a loadtime attribute.

rto_high
Specifies the TCP Retransmit Time out high value used in calculating factors and the maximum retransmits allowable used in TCP data segment retransmits. `rto_high` is the high factor. Default is 64. **rto_high** is a loadtime attribute.

sb_max
Specifies the maximum buffer size allowed for a socket. The default is 65,536 bytes. **sb_max** is a run attribute.

send_file_duration
Specifies the cache validation duration for all the file objects that system call send_file accessed in the Network Buffer Cache. This attribute is in number of seconds, the default is 300 for 5 minutes. 0 means that the cache will be validated for every access. This attribute only applies to AIX Version 4.3.2 or later.

site6_index
Specifies the maximum interface number for site local routing.

sockthresh
Specifies the maximum amount of network memory that can be allocated for sockets. 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** attribute represents a percentage of the **thewall** attribute, with possible values of 1 to 100 and a default of 85. **sockthresh** is a runtime attribute. This attribute only applies to AIX Version 4.3.1 or later.

somaxconn
Specifies the maximum listen backlog. The default is 1024 bytes. **somaxconn** is a runtime attribute. This attribute only applies to AIX Version 4.1.5, 4.2, or later.

subnetsarelocal
Determines if a packet address is on the local network. This attribute 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. **subnetsarelocal** is a runtime attribute.

tcp_ephemeral_low
Specifies the smallest port number to allocate for TCP ephemeral ports. The default is 32768. This attribute is available only in AIX 4.3.1 and beyond.

tcp_ephemeral_high
Specifies the largest port number to allocate for TCP

ephemeral ports. The default is 65535. This attribute is available only in AIX 4.3.1 and beyond.

tcp_keepidle

Specifies the length of time to keep the connection active, measured in half seconds. The default is 14,400 half seconds (7200 seconds or 2 hours). **tcp_keepidle** is a runtime attribute.

tcp_keepinit

Sets the initial timeout value for a tcp connection. This value is defined in 1/2 second units, and defaults to 150, which is 75 seconds. It can be changed to any value with the **−o** flag. **tcp_keepinit** is a runtime attribute.

tcp_keepintvl

Specifies the interval, measured in half seconds, between packets sent to validate the connection. The default is 150 half seconds (75 seconds). **tcp_keepintvl** is a runtime attribute.

tcp_mssdflt

Default maximum segment size used in communicating with remote networks. For AIX Version 4.2.1 or later, **tcp_mssdflt** is only used if path MTU discovery is not enabled or path MTU discovery fails to discovery a path MTU. **tcp_mssdflt** is a runtime attribute. The default value is 512. In AIX Version 4.3.3 and later versions, the **tcp_mssdflt** network option can also be set on a per interface basis via the **ifconfig** command.

tcp_ndebug

Specifies the number of **tcp_debug** structures. The default is 100. **tcp_ndebug** is a runtime attribute.

tcp_pmtu_discover

Enables or disables path MTU discovery for TCP applications. A value of 0 disables path MTU discovery for TCP applications, while a value of 1 enables it. The default value is 0. This attribute only applies to AIX Version 4.2.1 or later. **tcp_pmtu_discover** is a runtime attribute. In AIX Version 4.3.3 and later versions, the default value is 1 (enabled).

tcp_recvspace

Specifies the system default socket buffer size for receiving data. This affects the window size used by TCP. Setting the socket buffer size to 16KB (16,384) improves performance over Standard Ethernet and token−ring networks. The default is a value of 4096; however, a value of 16,384 is set automatically by the **rc.net** file or the **rc.bsdnet** file (if Berkeley−style configuration is issued).

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. In AIX Version 4.3.3 and later versions, the **tcp_recvspace** network option can also be set on a per interface basis via the **ifconfig** command.

The **tcp_recvspace** attribute must specify a socket buffer size less than or equal to the setting of the `sb_max` attribute. **tcp_recvspace** is a runtime attribute, but for daemons started by **inetd**, the following command needs to be executed:

```
 'stopsrc −s inetd ; startsrc −s inetd'
```

tcp_sendspace

Specifies the system default socket buffer size for sending data. This affects the window size used by TCP. Setting the socket buffer size to 16KB (16,384) improves performance over Standard Ethernet and Token−Ring networks. The default is a value of 4096; however, a value of 16,384 is set automatically by the **rc.net** file or the **rc.bsdnet** file (if Berkeley−style configuration is issued).

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
```

In AIX Version 4.3.3 and later versions, the **tcp_sendspace** network option can also be set on a per interface basis via the **ifconfig** command.

The `tcp_sendspace` attribute must specify a socket buffer size less than or equal to the setting of the `sb_max` attribute. **tcp_sendspace** is a runtime attribute, but for daemons started by **inetd**, the following command needs to be executed:

```
 'stopsrc −s inetd ; startsrc −s inetd'
```

tcp_timewait
The tcp_timewait option is used to configure how long connections are kept in the timewait state. It is given in 15 second intervals, and the default is 1.

tcp_ttl
Specifies the time to live for TCP packets. The default is 60 ticks (100 ticks per minute). **tcp_ttl** is a runtime attribute.

thewall
Specifies the maximum amount of memory, in kilobytes, that is allocated to the memory pool. In AIX Version 4.2.1 and earlier, the default value is 1/8 of real memory or 65536 (64 megabytes), whichever is smaller. In AIX Version 4.3, the default value is 1/8 of real memory or 131072 (128 megabytes), whichever is smaller. In AIX Version 4.3.1, the default value is 1/2 of real memory or 131072 (128 megabytes), whichever is smaller. In AIX Version 4.3.2 and later, the default value is 1/2 of real memory or 1048576 (1 gigabyte), whichever is smaller. **thewall** is a runtime attribute.

udp_ephemeral_low
Specifies the smallest port number to allocate for UDP ephemeral ports. The default is 32768. This attribute is available only in AIX 4.3.1 and beyond.

udp_ephemeral_high
Specifies the largest port number to allocate for UDP ephemeral ports. The default is 65535. This attribute is available only in AIX 4.3.1 and beyond.

udp_pmtu_discover
Enables or disables path MTU discovery for UDP applications. UDP applications must be specifically written to utilize path MTU discovery. A value of 0 disables the feature, while a value of 1 enables it. This attribute only applies to AIX Version 4.2.1 or later. **udp_pmtu_discover** is a runtime attribute. In versions prior to AIX Version 4.3.3, the default value is 0 (disabled); in AIX Version 4.3.3 and later versions, the default value is 1 (enabled).

udp_recvspace
Specifies the system default socket buffer size for receiving UDP data. The default is 41,600 bytes. The

`udp_recvspace` attribute must specify a socket buffer size less than or equal to the setting of the `sb_max` attribute. **udp_recvspace** is a runtime attribute.

udp_sendspace
Specifies the system default socket buffer size for sending UDP data. The default is 9216 bytes. The `udp_sendspace` attribute must specify a socket buffer size less than or equal to the setting of the `sb_max` attribute. **udp_sendspace** is a runtime attribute.

udp_ttl
Specifies the time to live for UDP packets. The default is 30 seconds. udp_ttl is a runtime attribute.

udpcksum
Allows UDP checksum to be turned on/off. A value of 0 turns it off; while a value of 1 turns it on. Default is 1. **udpcksum** is a runtime attribute.

> **Note:** If you use the `tcp_recvspace`, `tcp_sendspace`, `udp_recvspace` or `udp_sendspace` attribute to specify a socket to a buffer size larger than the `sb_max` attribute default, you must set the `sb_max` attribute to an equal or greater value. Otherwise, the socket system call returns the **ENOBUFS** error message when an application tries to create a socket.

use_isno
Enables the use of Interface Specific Network Options. The default value is 1 (enabled). This attribute only applies to AIX Version 4.3.3 and later versions.

Streams Tunable Attributes

The following Streams Tunable atrributes only apply to AIX Version 4.2 or later.

lowthresh
Specifies the maximum number of bytes that can be allocated using the **allocb** call for the BPRI_LO priority. When the total amount of memory allocated by the **net_malloc** call reaches this threshold, then the **allocb** request for the BPRI_LO prority returns 0. The **lowthresh** attribute represents a percentage of the **thewall** attribute and you can set its value from 0 to 100.

This is a runtime attribute and the default value is set to 90 (90% of **thewall** attribute).

medthresh
Specifies the maximum number of bytes that can be allocated using the **allocb** call for the BPRI_MED priority. When the total amount of memory allocated by the **net_malloc** call reaches this threshold, then the **allocb** request for the BPRI_MED prority returns 0. The **medthresh** attribute represents a percentage of the **thewall** attribute and you can set its value from 0 to 100.

This is a runtime attribute and the default value is set to 95 (95% of **thewall** attribute).

nstrpush
Specifies the maximum number (should be at least 8) of modules that you can push onto a single Stream.

This is a loadtime attribute and the default value is set to 8.

psebufcalls
Specifies the maximum number of bufcalls to allocate by Streams. The Stream subsystem allocates certain number of bufcall structures at initialization, so that when the **allocb** call fails, the user can register their requests for the **bufcall**. You are not allowed to lower this value until the system reboots, at which time it returns to

its default value.

This is a runtime attribute and the default value is set to 20.

pseintrstack Specifies the maximum size of the interrupt stack allowed by Streams while running in the offlevel. Sometimes, when a process running other than INTBASE level enters into a Stream, it encounters a stack overflow problem because the interrupt stack size is too small. Setting this attribute properly reduces the chances of stack overflow problems.

This is a loadtime attribute and the default value is set to 0x3000.

psetimers Specifies the maximum number of timers to allocate by Streams. The Stream subsystem allocates certain a number of timer structures at initialization, so that the streams driver or module can register their **timeout** calls. You are not allowed to lower this value until the system reboots, at which time it returns to its default value.

This is a runtime attribute and the default value is set to 20.

strctlsz 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). A **putmsg** call with a control part exceeding this size will fail with ERANGE.

This is a runtime attribute and the default value is set to 1024.

strmsgsz 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). 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.

This is a runtime attribute and the default value is set to 1024.

strthresh Specifies the maximum number of bytes Streams are normally allowed to allocate. When the threshold is passed, 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 (e.g. console continues to work properly). A value of zero means that there is no threshold.

The **strthresh** attribute represents a percentage of the **thewall** attribute and you can set its value from 0 to 100. The **thewall** attribute indicates the maximum number of bytes that can be allocated by Streams and Sockets using the **net_malloc** call. When you change **thewall** attribute, the threshold gets updated accordingly.

strturncnt Specifies the maximum number of requests handled by the current running thread for Module or Elsewhere level Streams synchronization.The Module level synchronization works in such a way that only one thread can run in the module at any time and all other threads which try to acquire the same module will enqueue their requests and leave. After the current running thread completes its work, it dequeues all the previously enqueued requests one by one and invokes them. If there are 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 the current running thread serves only the **strturncnt** number of threads, after that a separate kernel thread wakes up and invokes all the pending requests.

This is a runtime attribute and the default value is set to 15.

Examples

1. To change the maximum size of the mbuf pool to 3MB, enter:

   ```
   no −o thewall=3072
   ```

2. To reset the maximum size of the mbuf pool to its default size, enter:

   ```
   no −d thewall
   ```

3. To change the default socket buffer sizes on your system, add the following lines to the end of the **/etc/rc.net** file:

   ```
   /usr/sbin/no −o tcp_sendspace=16384
   /usr/sbin/no −o udp_recvspace=16384
   ```

4. To use an AIX machine as an internetwork router over TCP/IP networks, enter:

   ```
    no −o ipforwarding=1
   ```

Related Information

Network Overview for System Management in *AIX Version 4.3 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 Version 4.3 System Management Guide: Communications and Networks*.

Monitoring and Tuning Communications I/O in *AIX Versions 3.2 and 4 Performance Tuning Guide*.

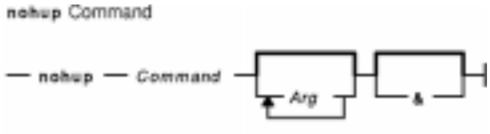
Path MTU Discovery in *AIX Version 4.3 System Management Guide: Communications and Networks*.

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.

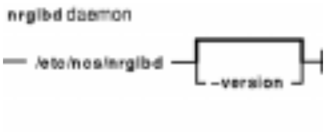
The **signal** subroutine.

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.

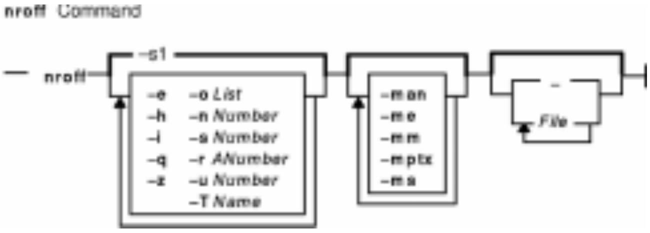
The Location Broker in *AIX Version 4.3 Communications Programming Concepts*.

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 AIX 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.

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.

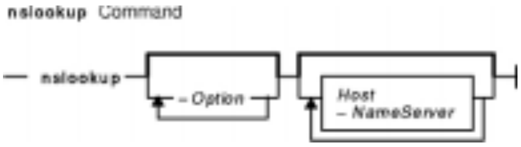
The article "nroff and troff Requests for the nroff and troff Commands" in the **troff** command.

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 **setroot** 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 **setsrchlist** command to specify a different list. Use the **setall** 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 **setall** 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 **setdomain** 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 **settimeout** 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 **setdomain** command. Use the **setall** 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.

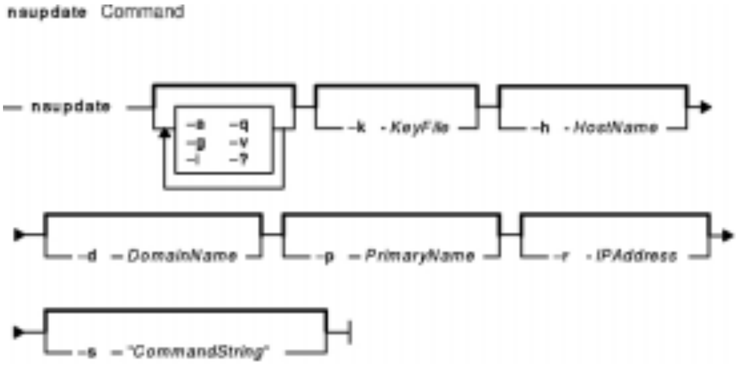
TCP/IP Name Resolution in *AIX Version 4.3 System Management Guide: Communications and Networks*.

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 Version 4.3 System Management Guide: Communications and Networks*.

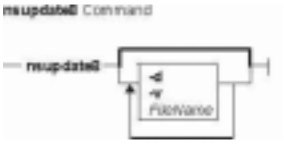
TCP/IP Daemons in *AIX Version 4.3 System Management Guide: Communications and Networks*.

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 **Prerequisite operations:**

are:

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 Version 4.3 System Management Guide: Communications and Networks*.

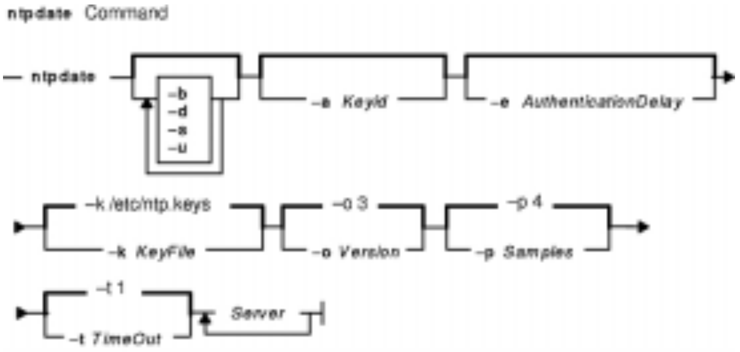
Configuring Name Servers and Planning for DOMAIN Name Resolution in *AIX Version 4.3 System Management Guide: Communications and Networks*.

ntpdate Command

Purpose

Sets the date and time using the Network Time Protocol (NTP). This command only applies to AIX Version 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**

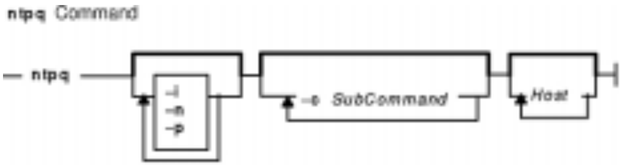
Daemons: **xntpd**,

ntpq Command

Purpose

Starts the standard Network Time Protocol (NTP) query program. This command only applies to AIX Version 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, enter:

   ```
   ntpq −i
   ```

2. To add a time interval of 1000 milliseconds to timestamps, enter:

   ```
   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.

authenticateyes | **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.

debugmore | **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 quieries 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.

hostnamesyes | **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.

ntpversion1 | **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:** Since **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 columns. The first colunm contains the index numbering the associations from 1 for internal use. The second column contains the actual association identifier returned by the server. The third column contains the status word for the peer. The rest of the 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 enter 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.

lassocations

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:

- address of the remote peer,
- reference ID (0.0.0.0 for an unknown reference ID),
- the stratum of the remote peer (a stratum of 16 indicates the remote peer is unsynchronized),
- type of the peer (local, unicast, multicast or broadcast),
- time the last packet was received, the polling interval (seconds),

- the polling interval (seconds),
- the reachability register (octal), and
- the 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 **hostanames no**.

Notes:

1. 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.

2. 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 enter 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 enter 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**

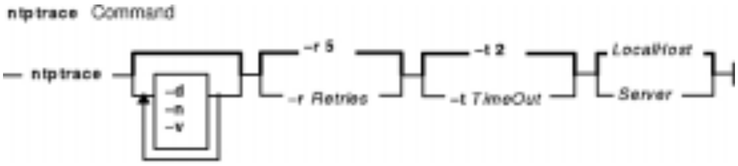
Daemons: **xntpd**,

ntptrace Command

Purpose

Traces a chain of Network Time Protocol (NTP) hosts back to their master time source. This command only applies to AIX Version 4.2 or later.

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**

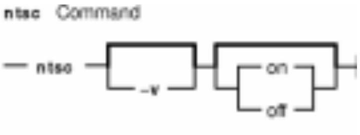
Daemons: **xntpd**,

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.

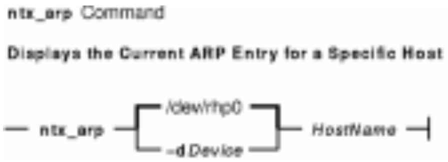
ntx_arp Command

Purpose

Displays and modifies the Network Terminal Accelerator (NTX) adapter's Internet−to−Ethernet address translation table.

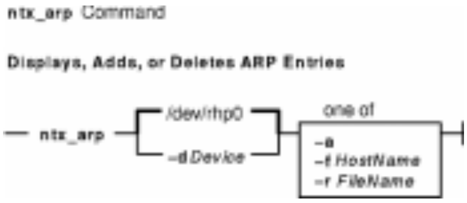
Syntax

To Display the Current ARP Entry for a Specific Host



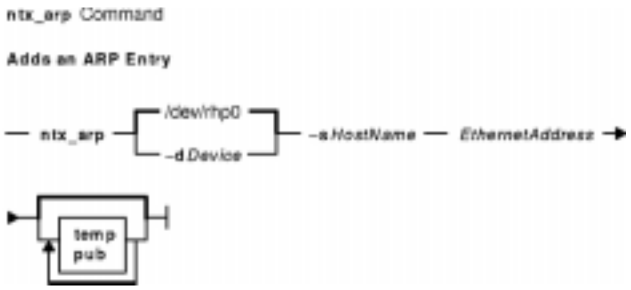
ntx_arp [**−d** *Device*] *HostName*

To Display, Add, or Delete ARP Entries



ntx_arp [**−d** Device] **−a** | **−r***HostName* | **−f***FileName*

To Add an ARP Entry



ntx_arp [**−d** *Device*] **−s** *HostNameEthernetAddress* [**temp**] [**pub**]

Description

The **ntx_arp** command displays and modifies the host adapter's Internet−to−Ethernet address translation table. The translation table is used by the address resolution protocol (ARP). By default, the **ntx_arp** command displays the current ARP entry for the host identified by the *HostName* parameter. You can specify a name or the Internet dot notation for the *HostName* parameter.

You can use the **ntx_arp** command to add an ARP entry for a host. To add an entry, use the **−s** flag and specify the *EthernetAddress* parameter. This parameter is given as 6 hexadecimal bytes separated by colons.

Entries in a table can be temporary, permanent, or published. By default, all entries are permanent. To add a temporary entry, specify the **temp** keyword with the **−s** flag. Temporary entries are maintained for 20 minutes. Published entries allow the host adapter to respond to ARP requests for a particular host even though the address is not the host adapter's address.

Flags

−a Displays the current ARP entries.

−d*Device* Specifies the raw device file name of the adapter. Use this option to override the default device name **/dev/rhp0**.

−f*FileName* Reads the specified file name and sets multiple entries in the ARP tables. Entries in the file should be of the form:

```
HostName EthernetAddress Type
```

These fields have the following values:

HostName
: Identifies the name of a host. You can specify a name or Internet dot notation.

EthernetAddress
: Specifies the host's Ethernet address. This is a field is 6 hexadecimal bytes separated by colons.

Type
: Defines the entry type. You can specify the **temp** or **pub** keyword for this field. If you do not specify either keyword, the default type permanent is assumed.

Entries are separated by new−line characters.

−r*HostName* Deletes an entry for the host specified by the *HostName* variable.

−s*HostName* Adds an ARP entry for the specified host name with the Ethernet address specified by the *EthernetAddress* parameter.

The Ethernet address is given as 6 hexadecimal bytes separated by colons. To add a temporary entry, specify the **temp** keyword with this flag. To specify a published entry, specify the **pub** keyword. If you do not specify a keyword, the command adds the entry as permanent.

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 display all ARP table entries for the second host adapter, enter:

```
ntx_arp −d /dev/rhp1 −a
```

Files

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

/dev/rhp0 Default NTX raw device file name.

Related Information

The **/dev/rhp** file.

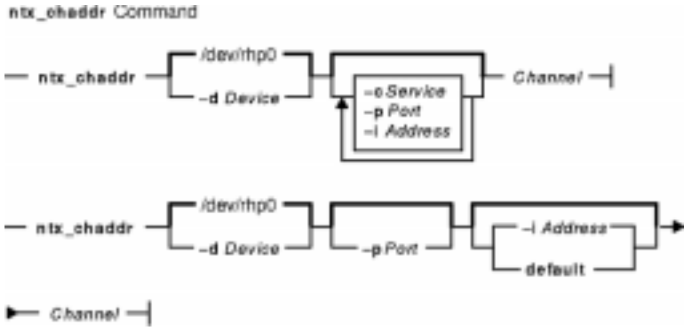
Network Terminal Accelerator Overview in *AIX Versions 3.2 and 4 Asynchronous Communications Guide*.

ntx_chaddr Command

Purpose

Displays the configurations of or configures individual Network Terminal Accelerator (NTX) channels.

Syntax



ntx_chaddr [**−d** *Device*] [**−c** *Service*] [**−p** *Port*] [**−i** *Address*] *Channel*

ntx_chaddr [**−d** *Device*] [**−p** *Port*] [**−i** *Address* | **default**] *Channel*

Description

The **ntx_chaddr** command displays the configuration of an adapter channel or configures an individual channel. To display the current configuration of a channel, use the **−d** flag with the *Channel* parameter. The *Channel* parameter can be a device file name or an adapter channel ID. This command can only access configured adapters.

Use the **ntx_chaddr** command to change the Internet address, protocol port, and call−type associated with an adapter channel. System administrators can use the **ntx_chaddr** command to identify groups of channels as call−out or call−in channels.

Before you can use the **ntx_chaddr** command, you must first configure both the adapter and its IP address. To configure the IP address, use the **ntx_ifcfg** command. To configure the adapter, use the **hty_load** command.

> **Note:** A channel ID is associated with a particular device. Channel IDs start at zero for each board. As a result, channel IDs do not always correspond to the system's minor device numbers.

The **ntx_chaddr** command updates one channel at a time. To configure multiple channels, you must issue the **ntx_chaddr** command for each channel.

Call−in Channels

A call−in channel accepts connections. The adapter accepts connections only to an adapter's assigned address and port.

The service that a call−in channel uses is determined by the destination port specified by the connection request. The connection request must come from one of three known terminal services: **telnet**, **rlogin**, or **nop**. To accept a connection to any port using a known terminal service, specify a 0 for the *Port* parameter. For more information about assigning services to a port, see the **ntx_pserv** command.

If you do not specify a *Port* parameter with the **−p** flag, the system uses the last port specified. If you do not specify an *Address* parameter with the **−i** flag, the system uses the last address specified. You can specify the **default** keyword instead of the **−i** flag to set the address to the system default. The system default address is assigned with the **ntx_ifcfg** command. The adapter automatically configures itself when a new local Internet address is specified.

Call−out Channels

A call−out channel requests a connection to a specific destination. To configure a call−out channel, specify the **−c** *Service* flag. You must specify the *Service* parameter. The valid services are **telnet**, **rlogin**, and **nop**.

When a host opens a call−out channel, the adapter attempts to establish a connection to the address and port specified with the **−i** and **−p** parameters. If the *Address* parameter is not specified, the command uses the last known value. If you do not specify the **−p** flag, the host database (typically **/etc/services**) is searched for the port assigned to the service you specified. If the port is not found, the **ntx_chaddr** command returns an error.

> **Note:** The service that a call−out channel uses is the determined by the *Service* parameter and not by the destination port.

Flags

−c*Service* Enables the channel as call−out. You must specify a *Service* value. Valid services are **telnet**, **rlogin**, and **nop**. You cannot use the **default** keyword with this flag.

−d*Device* Specifies the adapter raw device file name. If you do not specify a *Device* variable, the system uses the **/dev/rhp0** file. If the *Channel* parameter is a channel ID, the **−d** flag identifies the board corresponding to the ID. If the *Channel* parameter is a tty device file name, this flag can specify any valid adapter raw device file name.

−i*Address* Specifies an Internet address. For call−in channels, this flag specifies the address of the local adapter. For call−out channels, this flag specifies the remote address to connect to.

If you are configuring a call−in channel, you can use the **default** keyword instead of the **−i** flag to set the address back to the system default. You cannot use the **default** keyword with the **−c** flag.

−p*Port* Specifies a port number. For a call−in channel, this flag specifies the local protocol port. To accept connections to any port with terminal services, specify 0 for the *Port* parameter. For call−out channels, this flag specifies the remote port.

Parameters

Channel Specifies the device file name or an adapter channel ID.

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

Files

/usr/bin/ntx_chaddr	Contains the **ntx_chaddr** command.
/dev/rhp0	Default NTX raw device file name.

Related Information

The **hty_load** command, **ntx_ifcfg** command and, **ntx_pserv** commands.

The **/dev/rhp** file.

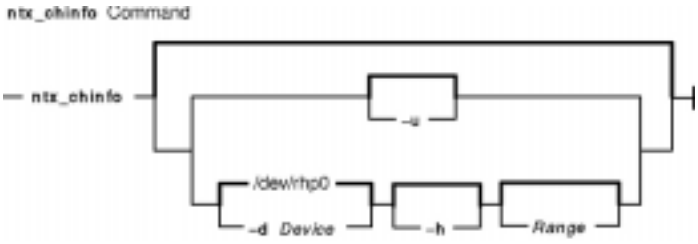
Network Terminal Accelerator Overview in *AIX Versions 3.2 and 4 Asynchronous Communications Guide*.

ntx_chinfo Command

Purpose

Displays Network Terminal Accelerator (NTX) channel information. This command only applies to AIX Version 4.2 or later.

Syntax



ntx_chinfo [**−u**] | [**−d** *Device*] [**−h**] [*Range*]

Description

The **ntx_chinfo** command displays network connection information for each channel on the specified channel range. The information contains the following parameters:

hty_no# The minor device number of the hty port corresponding to the channel.

chan# The channel number on the NTX board.

call_type The channel type, either CALL_IN or CALL_OUT.

service The service assigned when a channel is established.

client_address The inet address of the client holding network connection on the channel. If no network connection is held, then the value corresponds to the assigned call−out address .

name The name of the client holding the network connection.

state The state of the channel, can be one or several of the following values:

NET_CALIN
    A call−out channel is trying to establish network connection.

NET_CONN
    Channel has a network connection although either or both sides of the connection may have been lost. See NET_RDERR and NET_WRERR states.

NET_RECON
    Channel connection has been maintained across a close and open command sequence from the host.

NET_RDERR
    Network error on the input side of the connection.

NET_WRERR
    Network error on the output side of the connection.

TCS_CLOSI
    Channel is closing, waiting for the output data to drain.

TCS_CLSD

Channel closed by the host (may or may not yet have a network connection, see NET_CONN state).

TCS_OPEN

Channel opened by the host.

Flags

−d *Device* Specifies the raw device file name of the adapter. Use this option to override the default device name **/dev/rhp0**.

−h Displays header information showing date, device name, channel type, and total number of channels configured on the device.

−u Displays command usage.

Parameters

Range Specifies the channel range. You can enter this range in any of the following formats:

−EndChannel

Displays information from channel 0 to *EndChannel*.

StartChannel−

Displays information from channel *StartChannel* to the last configured channel id ont he NTX board.

StartChannel−EndChannel

Displays information from channel *StartChannel* to *EndChannel*.

If you do not enter *Range*, the entire range as configured on the board is used.

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 display network connection information for each channel from 0 to 15 on the /dev/rhp1 device, enter:

```
ntx_chinfo −d /dev/rhp1 −15
```

2. To display network connection information with a header for each channel from 200 to the last channel configured on the **/dev/rhp1** device, enter:

```
ntx_chinfo −d /dev/rhp1 −h 200−
```

3. To display network connection information for each channel from 100 to 115 on the **/dev/rhp0** device, enter:

```
    ntx_chinfo 100−115
```

Files

/usr/bin/ntx_chinfo	Contains the **ntx_chinfo** command.
/dev/rhp0	Default NTX raw device file name.

Related Information

The **ntx_chaddr**, and **ntx_pserv** commands.

The **/dev/rhp** file.

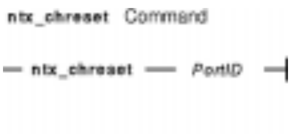
Network Terminal Accelerator Overview in *AIX Versions 3.2 and 4 Asynchronous Communications Guide*.

ntx_chreset Command

Purpose

Resets a Network Terminal Accelerator (NTX) channel. This command only applies to AIX Version 4.2 or later.

Syntax



ntx_chreset*PortID*

Description

The ntx_chreset command terminates any activity that is currently running on the channel specified by the hty port name. Any data buffered in the queues is flushed and the channel is closed. If the an **/etc/inittab** entry has the **getty** command scheduled to be respawned, it does so.

Flags

None.

Parameters

PortID Specifies the hty port name.

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 reset the channel with hty port name **/dev/hty123**, enter:

```
ntx_chreset /dev/hty123
```

Files

/usr/bin/ntx_chreset	Contains the **ntx_chreset** command.
/dev/rhp0	Default NTX raw device file name.

Related Information

The **ntx_reset** command.

The **/dev/rhp** file.

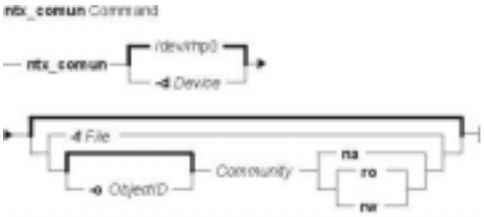
Network Terminal Accelerator Overview in *AIX Versions 3.2 and 4 Asynchronous Communications Guide*.

ntx_comun Command

Purpose

Displays and configures a Simple Network Management Protocol (SNMP) community profile for a Network Terminal Accelerator (NTX).

Syntax



ntx_comun [**−d** *Device*] [**−f** *File* | [**−o** *ObjectID*] *Community* { **na** | **ro** | **rw** }]

Description

The **ntx_comun** command displays and configures an adapter's SNMP community profile. A profile consists of community names and their respective permissions that determine access to Management Information Base (MIB) objects. If you do not specify any flags with the **ntx_comun** command, the command retrieves the current community configuration profile from the adapter and displays it on the standard output. The output is in the format required by the *File* parameter.

You can also use this command to configure community profiles. A community is a set of network management stations (NMSs) that have access to the adapter's SNMP agent. The *Community* parameter specifies the community name. The *Community* parameter is an ASCII string. You should configure community profiles with the **ntx_comun** command before issuing the **ntx_snmp** command to turn on the SNMP agent.

You use the **−o** flag to identify an ID for a MIB object or subtree. You can set the community's access mode to the object as one of the following:

na	not accessible
ro	read−only
rw	read−write

To configure permissions for multiple MIB subtrees for a single community, the host must issue the **ntx_comun** command multiple times for each subtree.

When the adapter's SNMP agent receives an protocol data unit (PDU), the agent first verifies that the community specified within the PDU has appropriate access to all of the MIB variables specified in the PDU. If the community profiles are not configured before the SNMP agent begins receiving PDUs, the SNMP agent uses a default read−only community profile with a community name of **public**.

Flags

−d*Device* Specifies the raw device file name of the adapter. Use this option to override the default device name **/dev/rhp0**.

−f *File* Reads the community profile configuration file and sends it to the adapter. A configuration file can have one or more entries with one entry per line. Each entry ends with a new−line character and has the following form:

ObjectID Community Mode

These fields have the following values:

ObjectID
: Identifies the MIB object. You can specify a single ID in dot notation or use the **all** keyword to specify all objects.

Community
: Specifies the community name.

Mode
: Sets the access mode. You can specify **ro** (read−only), **rw** (read/write), or **na** (not accessible).

Separate each field with a space or tab character. To create a comment line, start a line with a # (pound sign). Characters to the right of the # (pound−sign) are ignored.

−o*ObjectID* Identifies an MIB object. The *ObjectID* variable is a dot notation string. You can specify an ID for a leaf object or an MIB subtree. If the *ObjectID* variable identifies a subtree, access includes all elements of that tree. If the *ObjectID* variable is not specified, then access permission includes the whole MIB. Refer to the relevant MIB documents to determine the ID for an object.

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 set the community name to `public`, with access denied only to `1.3.6.1.2.1.1.3.0`, enter:

   ```
   ntx_comun −o 1.3.6.1.2.1.1.3.0 public na
   ```

2. To set the community with read−only access to the system group, enter:

   ```
   ntx_comun −o 1.3.6.1.2.1.1 rosys ro
   ```

3. To set the community with read−write access to the whole MIB, enter:

```
ntx_comun writer rw
```

4. To set the community with no access at all, enter:

```
ntx_comun criminal na
```

Files

/usr/bin/ntx_comun	Contains the **ntx_comun** command.
/dev/rhp0	Default NTX raw device file name.

Related Information

The **ntx_snmp** command.

The **rhp** file.

Network Terminal Accelerator Overview in *AIX Versions 3.2 and 4 Asynchronous Communications Guide*.

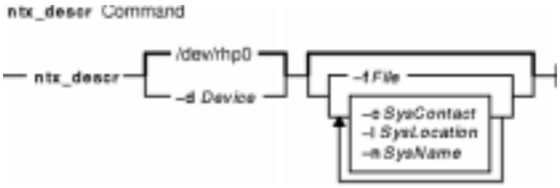
RFC 1213, Management Information Base for Network Management of TCP/IP−Based Internets: MIB−II.

ntx_descr Command

Purpose

Displays and assigns Simple Network Management Protocol (SNMP) site−specific variables for a Network Terminal Accelerator (NTX).

Syntax



ntx_descr [**−d** *Device*] [[**−f** *File*] | [[**−c** *SysContact*] [**−l** *SysLocation*]
[**−n** S*ysName*]]]

Description

The **ntx_descr** command displays and assigns site−specific values to SNMP Management Information Base (MIB) variables. The SNMP agent returns the site−specific values to the network management stations (NMS) when requested. If any value exceeds 79 bytes in length, it is silently truncated. More than one of the description variables can be specified on the command line.

If you do not specify any flags, the system reads the configuration from the adapter identified by the **/dev/rhp0** file and displays it on the standard output. The output is the same format as the *File* variable.

Flags

−c*SysContact* Identifies the contact person for an adapter. The *SysContact* variable is a string of any length. The default value is empty and of 0 length.

−d*Device* Specifies the adapter raw device file name. The default is the **/dev/rhp0** file.

−f *File* Reads the variable assignments from a file. Each line in the file is a single assignment of the following form:

Variable Value

The *Variable* field is one of the keywords, contact, location, or name. The second item is the *Value* of the variable. You can define a comment line with a # (pound sign). Everything to the right of the # (pound sign) is ignored.

−l*SysLocation* Specifies the physical location of a node (for example, `printer room`). The default value is empty and of length 0.

−n*SysName* Specifies the administratively assigned name for this managed node. By convention, this is the node's fully qualified domain name (for example, `smith@bld1_snmp.com`). The default value is empty and of length 0.

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

Files

/usr/bin/ntx_descr	Contains the **ntx_descr** command.
/dev/rhp0	Default NTX raw device file name.

Related Information

The **/dev/rhp** file.

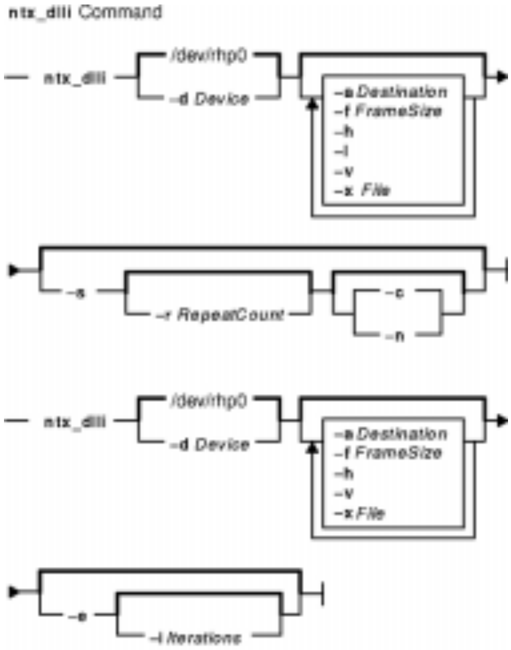
Network Terminal Accelerator Overview in *AIX Versions 3.2 and 4 Asynchronous Communications Guide*.

ntx_dlli Command

Purpose

Accesses the Diagnostic Link Layer Interface (DLLI) for the Network Terminal Accelerator (NTX).

Syntax



ntx_dlli [**−d** *Device*] [**−f** *FrameSize*] [**−h**] [**−l**] [**−v**] [**−s** [**−r** *RepeatCount*]
[**−c** | **−n**]] [**−a** *Destination*] [**−x** *File*]

ntx_dlli [**−d** *Device*] [**−f** *FrameSize*] [**−h**] [**−v**] [**−e** [**−i** *Iterations*]]
[**−a** *Destination*] [**−x** *File*]

Description

The **ntx_dlli** command accesses the DLLI and tests its functions. You can also use this command to generate spurious traffic on the network or saturate the network bandwidth for testing purposes.

> **Note:** The DLLI is a simple diagnostic interface only. Its message−passing interface cannot support higher−layer protocols above it and should not be used as a general−purpose interface to the link layer.

In normal mode, the **ntx_dlli** command transmits compiled−in link layer frames. Then the command enters an endless loop, receiving frames from the link layer and displaying them on the standard output device in raw hexadecimal format. To terminate the **ntx_dlli** command, use a **KILL** signal.

In external−loopback mode, the **ntx_dlli** command enters a loop and does the following:

- Transmits a compiled−in frame over the network port.

- Receives back the transmittal.
- Checks received frames against the transmitted frame for consistency.

If too many errors are recorded, the **ntx_dlli** command exits. Unless you specify the **−i** flag, this loop is endless. In external−loopback mode, an external−loopback connector is required on the AUI port. Use the **−v** flag: the command produces no output unless the **−v** flag is specified.

The **ntx_dlli** command uses a 64−byte compiled−in frame. To enlarge the frame up to 1024 bytes, use the **−f** flag.

Flags

−a*Destination* Identifies the destination address of the transmit frame. The *Destination* parameter must be a hexadecimal value (for example, `0:80:44:70:0:40`).

−c Specifies continuous transmission. Performs the DLLI transmit request endlessly until the command receives a **KILL** signal. This flag is only valid with the **−s** flag.

−d*Device* Sets the adapter raw−device file name. The default is the **/dev/rhp0** file.

−e Runs the command in external−loopback mode. This mode is used to continually drive the hardware. The loop continues until too many errors are received or the command receives a **KILL** signal. If you also specified the **−i** flag, the loop executes the number of times specified by the *Iteration* variable.

You cannot use the **−e** flag with the **−s** flag.

−f*FrameSize* Specifies a frame size other than the default 64 bytes. The maximum is 1024 bytes.

−h Gets the file descriptor for the adapter board from standard input. This is flag is only useful when running the **ntx_dlli** command via the **execve** subroutine. Because of the way the shell handles redirection, command lines of the type `ntx_dlli ... −h < device` will not work.

−i*Iterations* Specifies the number of iterations of the external−loopback mode. This flag is only valid with the **−e** flag.

−l Suppresses the display of frames on the standard output when the command is in the receive loop. Frames are received and then discarded before the command tries to receive another frame.

−n Exits, when the transmissions are complete, without entering the endless receive−frame loop. This flag is only valid when used in conjunction with the **−s** flag.

−r*RepeatCount* Specifies a repeat count. For each DLLI transmit request the host sends to the board using the **ntx_dlli** command, the command transmits the Ethernet frame the number of times specified by the *RepeatCount* variable. This option is useful for network loading tests. This flag is only valid with the **−s** flag.

−s Transmits a predefined, compiled−in frame on the network. The default is not to transmit anything and to enter the endless receive−frame loop. You cannot use this flag with the **−e** flag.

−v Specifies verbose output. At end of execution, the command prints the number of transmitted and received frames. Used with **−e** flag, the **−v** flag prints the result of each external loop.

−x *File* Identifies a frame to transmit. The *File* parameter must specify a file that contains a series of hex bytes separated by white space. The file does not have to contain as many bytes as the current frame size. Bytes that are not specified are set to zero.

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

Files

/usr/bin/ntx_dlli	Contains the **ntx_dlli** command.
/dev/rhp0	Default NTX raw device file name.

Related Information

The **/dev/rhp** file.

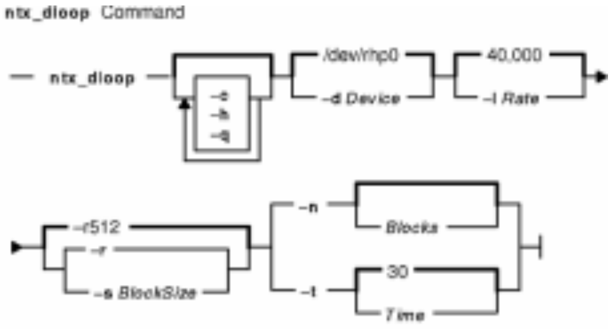
Network Terminal Accelerator Overview in *AIX Versions 3.2 and 4 Asynchronous Communications Guide*.

ntx_dloop Command

Purpose

Performs a direct memory access (DMA) loopback test for the Network Terminal Accelerator (NTX).

Syntax



ntx_dloop [**−c**] [**−q**] [**−d** *Device*] [**−h**] [**−l** *Rate*] { **−n** [*Blocks*] | **−t** [*Time*] }
[**−r** | **−s** *BlockSize*]

Description

> **Attention:** You must use the **kill** signal to stop this program. Do not use the **kill −9** command to close this program as the channel will not be closed properly.

The **ntx_dloop** command performs a DMA loopback test with the Network Terminal Accelerator diagnostic on an adapter. The loopback test does the following:

- Sends a block of data to the adapter.
- Reads the data from the adapter.
- Compares the data it reads against the original data.

Flags

−c Disable data verification. The application does not check that the data block read from the adapter matches the data block sent to adapter. The default is to verify the data.

−d*Device* Identifies the adapter raw device file name. By default, the **/dev/rhp0** file is used.

−h Gets the file descriptor for the adapter from standard input. This flag is only useful if you are running the command using the **execve** subroutine. As a result of the way the shell handles redirection, command lines of the type `ntx_dloop ... −h < /dev/rhp0` do not work.

−l*Rate* Sets the data throughput limit. You can use this flag to limit the use of the host adapter interface. Specify the *Rate* value in characters per section. The default value is 40,000.

−n*Blocks* Specifies the number of data blocks to send before terminating. By default, data blocks are sent for 30 seconds. You cannot use this flag with the **−t** flag.

−q Runs the command quietly. By default, the command reports test results to standard output. If you specify the **−q** flag, only errors are reported.

−r Varies the block sizes. When you specify this flag, the **ntx_dloop** command sends random blocks to the adapter. The block size varies from 1 to 512 blocks. You cannot use this flag with the **−s** flag. By default, the command uses a block size of 512.

−s *BlockSize* Fixes the block size. The *BlockSize* parameter specifies the size of the blocks to use with the command. You can specify any value between 1 and 512. You cannot use this flag with the **−r** flag.

−t *Time* Sets the program execution time. The *Time* variable specifies the length of time in seconds to loop before stopping. By default, the loop runs for 30 seconds. You cannot specify this flag with the **−n** flag.

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 run the DMA loopback on the default device for 3 minutes, do the following:

```
ntx_dloop −rt 180
```

This command runs using random block sizes and displays the result on standard output.

Files

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

/dev/rhp0 Default NTX raw device file name.

Related Information

The **/dev/rhp** file.

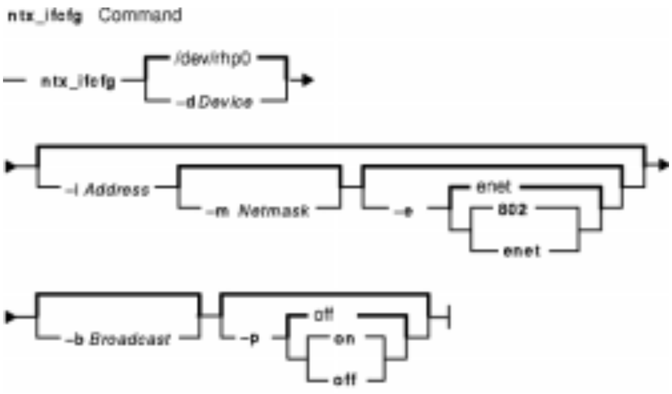
Network Terminal Accelerator Overview in *AIX Versions 3.2 and 4 Asynchronous Communications Guide*.

ntx_ifcfg Command

Purpose

Displays or assigns the Network Terminal Accelerator (NTX) network interface address.

Syntax



ntx_ifcfg [**−d***Device*] [**−i***Address* [**−m***Netmask*] [**−e** [**802** | **enet**]] [**−b***Broadcast*]
[**−p on** | **off**]

Description

The **ntx_ifcfg** command displays or assigns network interface addresses. If you do not specify any flags with this command, it displays the network address of the adapter identified by the **/dev/rhp0** file.

The **ntx_ifcfg** command is used to configure the adapter's Internet address immediately after the adapter is downloaded. No Internet network activity takes place until the Internet address is assigned. You can use this command after the adapter is loaded to redefine the board's Internet address.

Flags

−b*Broadcast*
Specifies the address that represents broadcasts to the network. By default, the *Broadcast* parameter is the address in which the host part appears as all ones.

−d*Device*
Sets the adapter raw device file name. The default is the **/dev/rhp0** file.

−e [**802** | **enet**]
Configures the adapter board for either IEEE 802.3 or an Ethernet operation. To place the board in 802 mode, specify the **802** keyword. To place the board in Ethernet mode, specify the **enet** keyword. The default mode is Ethernet.

−i*Address*
Specifies the Internet address of the adapter board. You can specify the network interface addresses by name or using Internet dot notation.

−m*Netmask* Reserves a portion of the address for subdividing the network into subnetworks. The mask includes the network part of the local address and the subnet part. The subnet is taken from the host field of the address.

You can specify the mask as a single hexadecimal number, with either a leading 0x (with a dot−notation Internet address) or with a network name. The mask contains ones for the bit positions in the 32−bit address used for the network and subnet parts. The mask contains zeros for the host part. The mask should contain at least the standard portion, and the subnet field should be contiguous with the network portion.

−p on | **off**

Turns promiscuous mode on or off. When promiscuous mode is on, the adapter acts as a network monitor and receives all well−formed frames, regardless of their destination address. The default is off.

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 Internet address for the default host adapter, enter:

```
ntx_ifcfg −i 192.6.153.99
```

Files

/usr/bin/ntx_ifcfg	Contains the **ntx_ifcfg** command.
/dev/rhp0	Default NTX raw device file name.

Related Information

The **/dev/rhp** file.

Network Terminal Accelerator Overview in *AIX Versions 3.2 and 4 Asynchronous Communications Guide*.

ntx_load Command

Purpose

Downloads code to a Network Terminal Acclerator (NTX) host adapter board.

Syntax

ntx_load Command
ntx_load
/dev/rhp0
-d Device
FileName

ntx_load [**−d***Device*] *FileName*

Description

> **Attention:** This command should not be run from the command line or invoked after system startup.

The **ntx_load** command downloads executable code to a host adapter and starts execution of the downloaded code. The download file must be in **.dnl** format and is specified by the *FileName* parameter.

Flags

−d*Device* Sets the adapter raw device file name. The default is the **/dev/rhp0** file.

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 download the Network Terminal Software to the default host adapter in the system, enter:

```
ntx_load /usr/lib/micrcode/ntx8023.04F.00
```

> **Note:** The adapter configuration methods call the **ntx_load** command.

Files

/usr/bin/ntx_load	Contains the **ntx_load** command.
/dev/rhp0	Default NTX raw device file name.

Related Information

The **/dev/rhp** file.

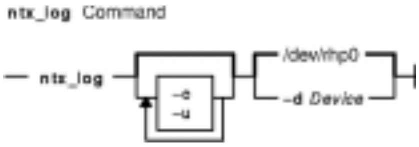
Network Terminal Accelerator Overview in *AIX Versions 3.2 and 4 Asynchronous Communications Guide*.

ntx_log Command

Purpose

Displays Network Terminal Accelerator (NTX) log information.

Syntax



ntx_log [**−c**] [**−u**] [**−d***Device*]

Description

The **ntx_log** command displays adapter log information from the log table on the adapter. By default, the information is formatted and displayed on standard output.

Flags

−c Clears the log table after it is read.

−d*Device* Sets the adapter raw device file name. The default is the **/dev/rhp0** file.

−u Writes the unformatted (binary) version of the log table to standard output. You can pipe this output into another program for interpretation.

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 write a formatted display of the default host adapter's log table and then clear the log table, enter:

```
ntx_log −c
```

Files

/usr/bin/ntx_log	Contains the **ntx_log** command.
/dev/rhp0	Default NTX raw device file name.

Related Information

The **/dev/rhp** file.

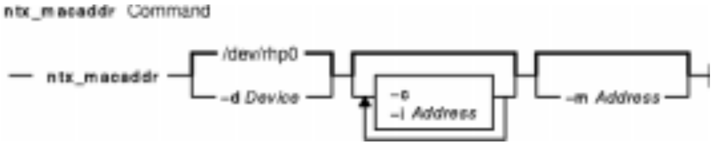
"Network Terminal Accelerator Overview in *AIX Versions 3.2 and 4 Asynchronous Communications Guide*.

ntx_macaddr Command

Purpose

Configures the Network Terminal Accelerator (NTX) MAC−layer addresses.

Syntax



ntx_macaddr [**−d***Device*] [**−i***Address*] [**−c**] [**−m***Address...*]

Description

The **ntx_macaddr** command configures the adapter MAC−layer addresses. By default, this command displays the current individual and multicast addresses of the adapter specified by the **/dev/rhp0** file. The addresses are sent to standard output. Addresses appear in hexadecimal notation (for example, `80:00:40:1a:22:f3`).

By default, the adapter is configured with a single individual MAC address. This address uniquely identifies the adapter. The address is factory−supplied and the **ntx_macaddr** command allows you to change it using the **−i** flag.

Flags

−c Clears the list of multicast addresses for which the adapter will receive frames.

−d*Device* Identifies the adapter raw device file. The **/dev/rhp0** file is the default.

−i*Address* Specifies an individual MAC−layer address. This is the address for the adapter's IEEE 802.3 MAC−layer address. The least significant bits of this address must be clear. For example, a valid address would be `fe:ff:ff:ff:ff:ff`.

−m*Address...* Adds a list of MAC−layer addresses as multicast addresses the adapter will receive. The least significant bits of these addresses must be set. You can only program a maximum of 32 multicast addresses at any one time.

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 add two multicast addresses, enter:

   ```
   ntx_macaddr −d /dev/rhp1 −m 1:0:0:0:0:0 83:0:4:1a:2:f3
   ```

2. To clear the existing multicast addresses, change an individual address, and add a new multicast address, enter:

   ```
   ntx_macaddr −c −i 0:80:0:70:0:1 −m 83:0:4:1a:2:00
   ```

Files

/usr/bin/ntx_macaddr	Contains the **ntx_macaddr** command.
/dev/rhp0	Default NTX raw device file name.

Related Information

The **/dev/rhp** file.

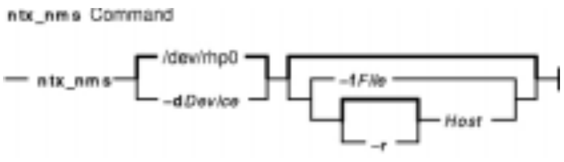
Network Terminal Accelerator Overview in *AIX Versions 3.2 and 4 Asynchronous Communications Guide*.

ntx_nms Command

Purpose

Displays and configures network management station (NMS) authentication for the Network Terminal Accelerator (NTX) Simple Network Management Protocol (SNMP) agent.

Syntax



ntx_nms [**−d***Device*] [**−f***File* | [**−r**] *Host*]

Description

The **ntx_nms** command displays and configures NMS authentication for the adapter's SNMP agent. If you issue this command without any flags, the command displays the NMS configuration for the adapter identified by the **/dev/rhp0** device file. The configuration is displayed on standard output in the same format required by the *File* parameter.

You configure NMS authentication by providing a list of authentication hosts to the SNMP agent. You can add to or remove from the list of authenticated hosts with the **ntx_nms** command. If you do not configure NMS authentication, then the SNMP agent accepts protocol data units (PDUs) from any NMS. To authenticate SNMP PDUs, run the **ntx_nms** command before starting the SNMP agent with the **ntx_snmp** command.

When authentication is in place, the SNMP agent discards PDUs from any NMS host that is not on its authentication list. If you have both NMS authentication and traps configured, the agent generates an authentication failure trap when it refuses a PDU. If traps are not configured, the agent discards the PDU and does not generate a trap. For information on configuring traps, see the **ntx_traps** command.

Flags

−d*Device* Specifies the adapter raw device file name. The default is the **/dev/rhp0** file.

−f*File* Identifies a NMS configuration file. This file contains a list of hosts to authenticate. Each entry in the list is separated by new−line characters. Each entry on the list is a host name or Internet address. Comment lines are designated with the # (pound sign). Characters to the right of the # (pound sign) are ignored.

−r *Host* Removes the specified host from the authentication list.

Parameters

Host Specifies a host name or an Internet address to add to the authentication list.

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 configure the SNMP agent to receive PDUs from the host at 192.6.153.6, enter:

```
ntx_nms 192.6.153.6
```

Files

/usr/bin/ntx_nms	Contains the **ntx_nms** command.
/dev/rhp0	Default NTX raw device file name.

Related Information

The **ntx_snmp**, and **ntx_traps** commands.

The **/dev/rhp** file.

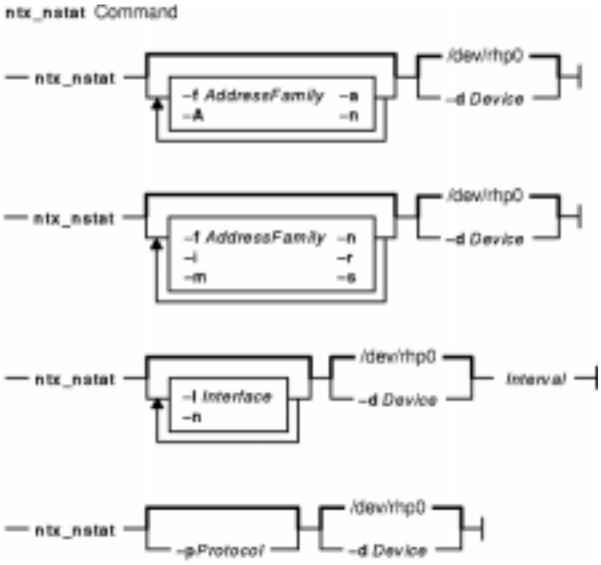
Network Terminal Accelerator Overview in *AIX Versions 3.2 and 4 Asynchronous Communications Guide*.

ntx_nstat Command

Purpose

Displays network information for the Network Terminal Accelerator (NTX).

Syntax



ntx_nstat [**−d***Device*] [**−A**] [**−a**] [**−n**] [**−f***AddressFamily*]

ntx_nstat [**−d***Device*] [**−i**] [**−m**] [**−n**] [**−r**] [**−s**] [**−f***AddressFamily*]

ntx_nstat [**−d***Device*] [**−n**] [**−I** *Interface*] *Interval*

ntx_nstat [**−d***Device*] [**−p***Protocol*]

Description

The **ntx_nstat** command displays the contents of various network−related data structures known to the Network Terminal Accelerator adapter. You can use this command to display the following information:

- A list of active sockets for each protocol
- The contents of network data structures
- Information regarding packet traffic on the configured network interfaces
- Statistics about a named protocol

You can use the **−n** flag with any of the display formats to view network addresses in dot decimal form.

Active Sockets

To display information about active sockets use, the **−a** flag. This flag displays the following information:

- Local addresses
- Remote addresses
- Send queue sizes in bytes
- Receive queue sizes in bytes
- Protocol
- Internal state of the protocol

If a socket's address specifies a network, but no specific host address, the address appears in the form `host.port` or `network.port`. If a host or network address is to the adapter's host machine, the name of the host or network is displayed as it appears in the **/etc/hosts** or **/etc/networks** file. If a host or network is unknown, the **ntx_nstat** command displays the Internet address instead of the name. The format of the numeric address depends on the address family. Unspecified or wildcard address and ports appear as an * (asterisk).

To display the address of any protocol control blocks associated with sockets, use the **−A** flag with the **−a** flag.

Network Data Structures

To view information about the adapter's interfaces, use the **−i** flag. Currently, the adapter installation automatically configures the adapter's Ethernet address and the adapter's loopback interface.

Routing Status

To display the routing tables, use the **−r** flag. The routing table tracks the available routes and their statuses. Each route consists of a destination host or network and a gateway to use in forwarding packets. The routing table contains the following fields:

```
Destination   Gateway   Flags   Refs   Use   Interface
```

These fields contain the following information:

`Destination` Identifies the destination host or network.

`Gateway` Identifies the address of the outgoing interface. Direct routes are created for each interface attached to the local host.

`Flags` Displays the state of the route. A route can have the following states:

U
: The route is up.

G
: The route is a gateway.

D
: The route was created dynamically by a redirect.

M
: The route was modified by a redirect.

`Refs` Identifies the current number of active uses of the route. Connection−oriented protocols normally hold on to a single route for the duration of a connection. Connectionless protocols obtain a route while sending to the same destination.

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

`Interface` Indicates the network interface utilized for a route.

To display routing statistics instead of the routing table, use the **−s** flag with the **−r** flag.

Network Interface Statistics

To display a running count of the statistics related to the primary network interface and summary statistics for all other interface activity, use the *Interval* parameter. The primary network interface is the first interface found during autoconfiguration. The first line output summarizes system activity since the last reboot. Subsequent lines of output show values accumulated over the preceding interval.

To view information for a single interface, use the **−I** flag. This type of display provides a table of cumulative statistics for packets transferred, errors, and collisions. The network addresses of the interface and the maximum transmission unit (MTU) are also displayed.

Flags

−A Displays the status of protocol control blocks associated with sockets. This flag is used for debugging. You must use this flag with the **−a** flag.

−a Displays the state of all sockets. Normally, sockets used by server processes are not shown.

−d*Device* Specifies the adapter device file. The default is the **/dev/rhp0** file.

−f*AddressFamily* Limits statistics or address control−block reports to those specified by the *AddressFamily* parameter. The only valid address family is **inet** (**AF_INET**).

−i Displays the state of autoconfigured interfaces. Interfaces statically configured into a system, but not located at boot−time, are not shown.

−I*Interface* Displays information about the interface identified by the *Interface* parameter. This flag is used with the *Interval* parameter.

−m Displays statistics recorded by the adapter memory management routines including the allocation of memory buffers (mbufs).

−n Displays Internet addresses instead of host or network names. You can use this flag with any of the display formats.

−p*Protocol* Displays statistics about the specified protocol. You can specify either a well−known name or an alias for the *Protocol* parameter. Some protocol names and aliases are listed in the **/etc/protocols** file. A null response typically means that there are no statistics to report. If the protocol is unknown or if there is no statistics routine for it, the program displays the following message:

```
ntx_nstat: unknown or uninstrumented protocol
```

−r Displays the routing tables. When the **−s** flag is also present, the **−r** flag shows routing statistics instead.

−s Displays per−protocol statistics.

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/bin/ntx_nstat	Contains the **ntx_nstat** command.
/dev/rhp0	Default NTX raw device file name.

Related Information

The **/dev/rhp** file.

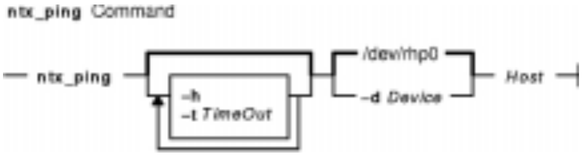
Network Terminal Accelerator Overview in *AIX Versions 3.2 and 4 Asynchronous Communications Guide*.

ntx_ping Command

Purpose

Sends an echo request to a network host from the Network Terminal Accelerator (NTX).

Syntax



ntx_ping [**−d***Device*] [**−h**] [**−t***TimeOut*] *Host*

Description

> **Attention:** You should stop this program with a **kill** signal. Do not use the **kill −9** command since the command may not properly close the diagnostic channel.

The **ntx_ping** command sends an echo request to a network host. The command uses the Network Terminal Accelerator diagnostic to send an Internet control message protocol (ICMP) ECHO_REQUEST datagram to a host. The command expects an ECHO_RESPONSE datagram in return. The datagrams are 64 bytes in length. If the host responds within the specified timeout (specified in seconds), the **ntx_ping** command prints `host is alive`. Otherwise it prints `no answer from host`.

Flags

−d*Device* Specifies the adapter raw−device file name. The default is the **/dev/rhp0** file.

−h Retrieves the file descriptor for the adapter from standard input. This flag is only useful when running the **ntx_ping** command via the **execve** subroutine. Because of the way the shell handles redirection, command lines of the type `ntx_ping ... −h hostname < device` do not work.

−t*TimeOut* Specifies the time to wait, in seconds, for a response. The default is 20 seconds.

Parameters

Host Specifies a host name or an Internet address.

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

To send a **ping** command echo request to a host named `fred`, enter:

```
ntx_ping fred
```

Files

/usr/bin/ntx_	Contains the **ntx_** command.
/dev/rhp0	Default NTX raw device file name.

Related Information

The **/dev/rhp** file.

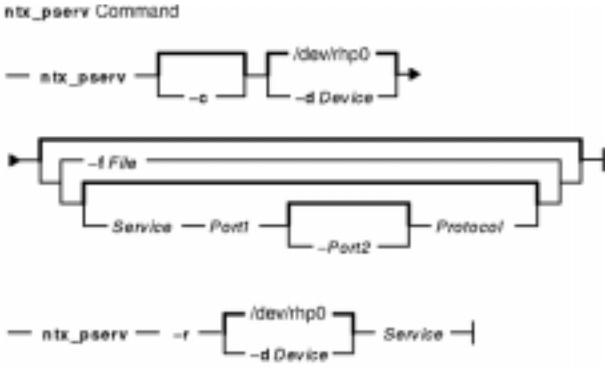
Network Terminal Accelerator Overview in *AIX Versions 3.2 and 4 Asynchronous Communications Guide*.

ntx_pserv Command

Purpose

Displays and assigns services to protocol ports for the Network Terminal Accelerator (NTX).

Syntax



ntx_pserv [**−c**] [**−d***Device*] [**−f***File* | [*ServicePort1* [*−Port2*] *Protocol*]]

ntx_pserv−r [**−d** *Device*] *Service*

Description

The **ntx_pserv** command displays assigned services and assigns services to a protocol ports. Once a protocol port is assigned a service, you cannot assign it to a different service without clearing all entries for the service with the **−r** flag or clearing the entire database with the **−c** flag. You can assign a port to services:

- **telnet**
- **rlogin**
- **nop**
- **snmp**
- **snmp−trap**

The **nop** (no−operation) service is a transparent terminal service that provides no special processing.

You can assign a service to a single port a range of ports. To specify a single port, use the *Port1* parameter. To specify a range of ports, use the *Port1* and *Port2* parameters, separated by a − (minus sign). You can set the *Protocol* parameter to **tcp** or **udp**.

You can change a service assignment at any time. However, some changes do not take immediate effect. The **telnet**, **rlogin**, and **nop** services affect the next connection request a port receives. The **snmp** and **snmp−trap** services have an affect only before the adapter has been enabled with the **ntx_ifcfg** command.

By default, the **ntx_pserv** command displays the current service assignments for the adapter identified by the **/dev/rhp0** file.

Flags

−c Clears all entries for all services, ports, and protocols. The default is to add the new services to the current configuration. If the **−f** flag or a service is also specified, the **ntx_pserv** command clears the database before adding new services.

−d*Device* Specifies the adapter raw device file name. The **/dev/rhp0** file is the default.

−f*File* Reads the service configuration from the *File* parameter and sends it to the adapter. The service specification is in the same format as the command line:

```
Service Port1[−Port2] Protocol
```

Each service specification is separated from the next with a new line. The # (pound sign) identifies a comment line. Everything to the right of the # is ignored.

−r*Service* Removes all entries for the specified service on all ports and protocols.

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

Files

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

/dev/rhp0 Default NTX raw device file name.

Related Information

The **ntx_ifcfg** command.

The **/dev/rhp** file.

Network Terminal Accelerator Overview in *AIX Versions 3.2 and 4 Asynchronous Communications Guide*.

ntx_reset Command

Purpose

Resets the Network Terminal Accelerator (NTX) adapter. This command only applies to AIX Version 4.2 or later.

Syntax

ntx_reset Command
ntx_reset
/dev/rhp0
-d Device

ntx_reset [**−d** *Device*]

Description

The **ntx_reset** command issues a hardware reset command to the NTX host adapter board.

> **Attention:** Use this command with great caution. You should stop all activity on the NTX board and have no processes waiting on the board when issuing this command. For example, hty ports must not be login enabled.

Flags

−d *Device* Specifies the raw device file name of the adapter. Use this option to override the default device name **/dev/rhp0**.

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 reset the default NTX host adapter, enter:

```
ntx_reset
```

Files

/usr/bin/ntx_reset	Contains the **ntx_reset** command.
/dev/rhp0	Default NTX raw device file name.

Related Information

The **/dev/rhp** file.

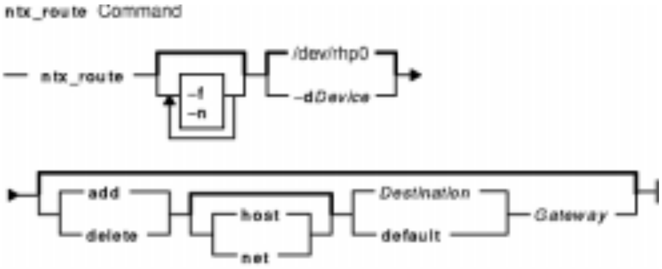
Network Terminal Accelerator Overview in *AIX Versions 3.2 and 4 Asynchronous Communications Guide*.

ntx_route Command

Purpose

Manipulates the Network Terminal Accelerator (NTX) routing tables.

Syntax



ntx_route [**−f**] [**−n**] [**−d***Device*] [{ **add** | **delete** } [**host** | **net**]
{ *Destination* | **default** } *Gateway*]

Description

The **ntx_route** command manipulates the network routing tables on the host adapter. You can use this command to:

- Flush the routing tables of all gateway entries.
- Add a route to a host or network.
- Delete a route from a host or network.

When using the **ntx_route** command, you must distinguish between routes to a host from routes to a network. To identify the route destination as a network, specify the **net** keyword. To identify a route destination as a host, specify the **host** keyword. If you do not specify a keyword, the **ntx_route** command uses the *Destination* parameter to determine the type of route. If the *Destination* parameter contains a local address that is part of INADDR_ANY, the command assumes the destination is network. Otherwise, the command assumes the destination is a host.

The keyword **default** is available for gateways to use after all other routes have been attempted (the destination is set to INADDR_ANY).

Flags

−d Sets the adapter raw device file name. The default is the **/dev/rhp0** file.

−f Flushes the routing tables of all gateway entries. If this flag is used in conjunction with one of the other flags or keywords, the **ntx_route** command flushes the gateways before performing other actions.

−n Displays host and network names as Internet addresses. The default is to display the simple host or network name.

Parameters

Destination Specifies the destination host or network.
Gateway Specifies the network gateway to which packets should be addressed.

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 add destination to the routing table, enter:

```
ntx_route add Destination Gateway
```

Files

/usr/bin/ntx_route Contains the **ntx_route** command.
/dev/rhp0 Default NTX raw device file name.

Related Information

The **/dev/rhp** file.

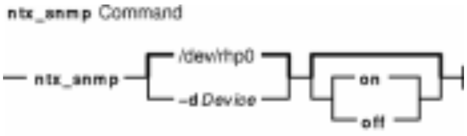
Network Terminal Accelerator Overview in *AIX Versions 3.2 and 4 Asynchronous Communications Guide*.

ntx_snmp Command

Purpose

Sets and displays the state of the Network Terminal Accelerator (NTX) Simple Network Management Protocol (SNMP) agent.

Syntax



ntx_snmp [**−d***Device*] [**on** | **off**]

Description

The **ntx_snmp** command sets and displays the state of the adapter's SNMP agent. You use this command to turn the SNMP agent on or off.

Before turning the SNMP agent on, consider issuing the **ntx_comun** and **ntx_nms** commands. The **ntx_comun** command configures network management station (NMS) community profiles. A community profile is a set of permissions for access to the SNMP agent's Management Information Base (MIB). The **ntx_nms** command determines which NMSs are authorized to interact with the SNMP agent.

If the cold start trap is enabled, the agent sends the trap protocol data unit (PDU) when the first **ntx_snmp on** command is issued. For information about enabling the cold start trap and other traps, see the **ntx_traps** command.

To stop an SNMP agent, issue the **ntx_snmp off** command. Stopping the agent may be necessary if a configuration error occurs, if an NMS malfunctions, or if a security breach is detected. After the network administrator has remedied the problem, the host can restart the SNMP agent by issuing the **ntx_snmp on** command.

SNMP Agent

The SNMP implementation consists of an SNMP agent that executes on intelligent communications adapters. The SNMP agent manages the adapter and its associated software. The SNMP agent does not manage the host computer.

Once configured, the SNMP agent responds to PDUs received from the network. The host utilities provide a means for a network administrator to configure the SNMP agent with site−specific information. For detailed information on the individual host utilities, see *RFC 1157, Simple Network Management Protocol* and *RFC 1213, Management Information Base for Network Management of TCP/IP−Based Internets: MIB−II*.

> **Note:** The SNMP agent complies with both MIB−II (RFC 1213) and the SNMP protocol (RFC 1157).

The host must explicitly start the on−board agent with the **ntx_snmpon** command. Before the host issues the **ntx_snmp on** command, the SNMP agent does not respond to any SNMP PDUs. This allows the host to configure community profiles and to configure which NMSs have access to the agent.

SNMP Profiles

The **ntx_nms** and **ntx_comun** commands configure the NMSs and community profiles, respectively. If the network administrator does not require this security, the SNMP agent uses a default read−only community profile. In order to use the **public** default community profile, the host may issue the **ntx_snmp on** command without issuing the other commands. This profile allows the **public** community read−only access to the entire MIB.

Along with starting and stopping the SNMP agent and configuring community profiles, the host can configure some site−specific information which the agent holds in its MIB. See the **ntx_descr** command for more information about configuring site−specific information.

Flags

−d*Device* Sets the adapter raw device file name. The default is the **/dev/rhp0** file.

Exit Status

This command returns the following exit values:

0 Indicates the agent is on if neither the **on** nor **off** option was specified with the command. If either the **on** or **off** keyword was specified, a 0 indicates success.

1 Indicates the agent is off.

2 Indicates a failure.

Security

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

Auditing Events: N/A

Examples

1. To turn on the SNMP agent, enter:

   ```
   ntx_snmp on
   ```

2. To get the current state of the agent, enter:

   ```
   ntx_snmp
   SNMP is on
   ```

Files

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

/dev/rhp0 Default NTX raw device file name.

Related Information

The **ntx_comun** command, **ntx_descr** command, **ntx_nms** command and, **ntx_traps** command.

The **/dev/rhp** file.

Network Terminal Accelerator Overview in *AIX Versions 3.2 and 4 Asynchronous Communications Guide*.

RFC 1157, Simple Network Management Protocol.

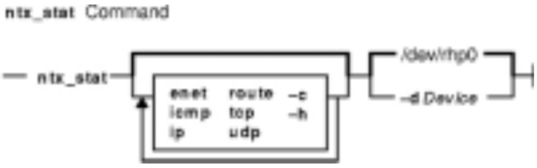
RFC 1213, Management Information Base for Network Management of TCP/IP−Based Internets: MIB−II.

ntx_stat Command

Purpose

Displays and clears Network Terminal Accelerator (NTX) diagnostic information.

Syntax



ntx_stat [**−c**] [**−d***Device*] [**−h**] [**enet**] [**icmp**] [**ip**] [**route**] [**tcp**] [**udp**]

Description

The **ntx_stat** command displays and clears adapter diagnostic information. This command retrieves, and optionally clears, statistics from various software modules on the adapter. To retrieve particular statistics, specify one of the following keywords:

enet Ethernet statistics

icmp Internet Control Message Protocol (ICMP) statistics

ip Internet Protocol (IP) statistics

route Route statistics

tcp Transmission Control Protocol (TCP) statistics

udp User Datagram Protocol (UDP) statistics

The **ntx_stat** command displays the selected statistics on the standard output device in a user−friendly display. If no statistics parameters are given, the command displays all the statistics.

Flags

−c Clears the statistics after they are retrieved.

−d*Device* Sets the adapter raw device file name. The default is the **/dev/rhp0** file.

−h Gets the file descriptor for the adapter board from standard input. This flag is only useful when running the **ntx_stat** command via the **execve** subroutine. Because of the way the shell handles redirection, command lines of the type `ntx_stat ... −h < device` do not work.

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

To retrieve the Ethernet and TCP statistics and clear them, enter:

```
ntx_stat −c enet tcp
```

Files

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

/dev/rhp0 Default NTX raw device file name.

Related Information

The **/dev/rhp** file.

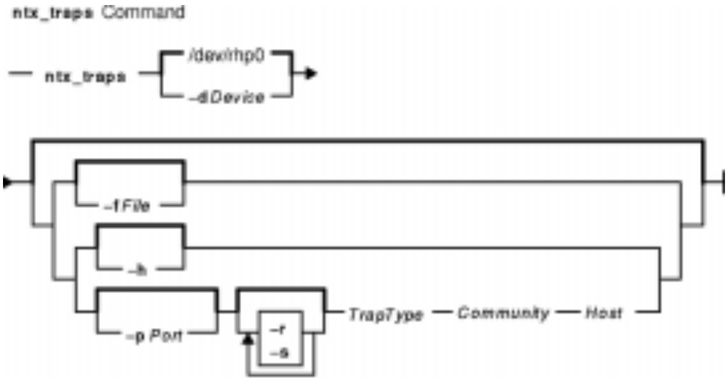
Network Terminal Accelerator Overview in *AIX Versions 3.2 and 4 Asynchronous Communications Guide*.

ntx_traps Command

Purpose

Displays or configures the Network Terminal Accelerator (NTX) Simple Network Management Protocol (SNMP) traps.

Syntax



ntx_traps [**−d***Device*] [[**−f***File*] | [**−h**] | [[**−p***Port*] [**−r**] [**−s**] *TrapType*
Community Host]]

Description

The **ntx_traps** command displays or configures the adapter SNMP traps. A trap is an unsolicited SNMP message that the agent transmits when specific events occur. By default, this command displays the current trap configuration for the adapter identified by the **/dev/rhp0** file. The configuration is sent to standard output in the format required by the *File* parameter.

The **ntx_traps** command determines which trap protocol data units (PDUs) the SNMP agent sends to which network management station (NMS). If the host does not issue the **ntx_traps** command, the agent does not transmit trap PDUs. IBM supports the standard SNMP generic traps as well as some enterprise−specific traps that are unique to the adapter.

> **Note:** It is possible, and simple, for the host to configure more than one NMS to receive the same trap type.

The *TrapType* parameter is a decimal number or symbol that specifies the SNMP trap type. A trap can be generic or specific. If the **−s** flag is specified, the *TrapType* parameter is interpreted as a specific trap. Otherwise, it is a generic trap. Valid values for trap types are integers or the corresponding symbolic names. Valid generic trap types are:

- **0** (**coldstart**)
- **1** (**warmstart**)
- **2** (**linkdown**)
- **3** (**linkup**)
- **4** (**authfail**)
- **5** (**egploss**)

The SNMP agent does not generate **warmstart**, **linkdown**, **linkup**, or **egploss** traps.

Specific traps identify enterprise−specific traps. Refer to the **/usr/include/ntx/st.snmp.h** file for the current list of enterprise−specific traps. The following traps are defined:

0 (**hostrej**) Connection attempt rejected by host computer.

1 (**netrej**) Connection attempt rejected by terminal.

2 (**neterr**) Unrecoverable network error occurred on the session.

Flags

Community Specifies the community to generate the trap to. The parameter is normally an ASCII string that serves to identify a set of NMSs that have access to the adapter's SNMP agent. The SNMP agent uses the community specified in the PDU when generating the specified trap to the specified NMS.

−d*Device* Sets the adapter raw device file name. The default is the **/dev/rhp0** file.

−f*File* Identifies a trap configuration file. The file contains an entry for each port to configure. Entries are separated by new−line characters and have the following form:

```
Port GenericTrap SpecificTrap Community Host
```

These fields have the following values:

Port
UDP port number.

GenericTrap
Type of generic trap.

SpecificTrap
Type of specific trap.

Community
Name of the community.

Host
Name of the host or Internet address of the host.

Fields in an entry are separated by blank space. The # (pound sign) indicates a comment line. Characters to the right of a # (pound sign) are ignored.

Host Specifies the Internet address or host name of the NMS that is to receive the specified trap PDU.

−h Provides help on symbolic names for traps.

−s Interprets the *TrapType* parameter as a specific trap.

−r Disables the SNMP agent from generating the specified trap to the specified NMS.

−p*Port* Specifies the UDP port number that the NMS uses to receive the trap. The default port is 162.

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 configure cold start traps to be sent to hosts 192.6.153.6 and 192.6.153.2, enter:

   ```
   ntx_traps 0 public 192.6.153.6
   ntx_traps 0 public 192.6.153.2
   ```

2. To configure authentication failure traps to be sent to host 192.6.153.6, enter:

   ```
   ntx_traps authfail private 192.6.153.6
   ```

3. To configure session network error traps to be sent to host 192.6.153.20, enter:

   ```
   ntx_traps −s neterr public 192.6.153.20
   ```

Files

/usr/bin/ntx_traps	Contains the **ntx_traps** command.
/dev/rhp0	Default NTX raw device file name.

Related Information

The **/dev/rhp** file.

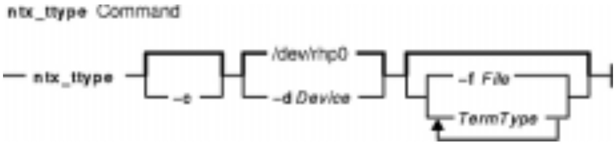
Network Terminal Accelerator Overview in *AIX Versions 3.2 and 4 Asynchronous Communications Guide*.

ntx_ttype Command

Purpose

Adds to, displays, or clears the terminal type database for the Network Terminal Accelerator (NTX).

Syntax



ntx_ttype [**−c**] [**−d***Device*] [[**−f***File*] | [*TermType* ...]]

Description

The **ntx_ttype** command adds to, displays, and clears the terminal type database. By default, this command displays the terminal type database for the adapter identified by the **/dev/rhp0** file. You can also use this command to add one or more entries to the terminal type database.

The terminal type database is only used by the **telnet** protocol service to find terminal types that are known to the host. The adapter requests terminal types from the remote database until a match is found or the remote runs out of terminal types it can emulate. If a match is found, the adapter uses the name in the database. For example, if `ANSI` is received, it matches `ansi` in the database.

To add a single entry, specify the *TermType* parameter. To add several entries, specify the **−f***File* flag. Each entry in the database is limited to 40 characters and must contain only ASCII characters. The command ignores duplicate entries. The entries are case−insensitive so that, for example, the entry `ansi` is not added if `ANSI` is already in the database. You can change this database at any time.

Flags

−c Clears database entries. By default, the command adds entries.

−d*Device* Sets the adapter raw device file name. The **/dev/rhp0** file is the default.

−f*File* Identifies a file with a list of terminal types. Each entry is listed on a separate line. The # (pound sign) is used to start a comment. Everything to the right of the # (pound sign) is ignored.

Parameters

TermType ... Specifies the terminal types.

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

Files

/usr/bin/ntx_ttype	Contains the **ntx_ttype** command.
/dev/rhp0	Default NTX raw device file name.

Related Information

The **/dev/rhp** file.

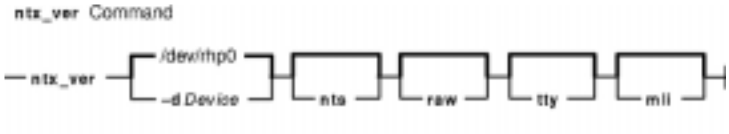
Network Terminal Accelerator Overview in *AIX Versions 3.2 and 4 Asynchronous Communications Guide*.

ntx_ver Command

Purpose

Displays Network Terminal Accelerator (NTX) version information.

Syntax



ntx_ver [**−d***Device*] [**nts**] [**raw**] [**tty**] [**mli**]

Description

The **ntx_ver** command displays the adapter version information from the host drivers and the Network Terminal Software (NTS). You can specify the following keywords with this command:

nts	Network Terminal Software version information
raw	Raw driver version information
tty	tty driver version information
mli	MLI driver version information

If you do not specify a keyword, all the version information is displayed.

Flags

−d*Device* Sets the adapter raw device file name. The **/dev/rhp0** file is the default.

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

To retrieve the raw and tty driver version information, enter:

```
ntx_ver raw tty
```

Files

/usr/bin/ntx_ver	Contains the **ntx_ver** command.
/dev/rhp0	Default NTX raw device file name.

Related Information

The **/dev/rhp** file.

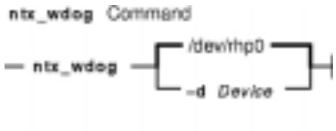
Network Terminal Accelerator Overview in *AIX Versions 3.2 and 4 Asynchronous Communications Guide*.

ntx_wdog Command

Purpose

Starts the watchdog timer on a Network Terminal Accelerator (NTX) adapter board. This command only applies to AIX Version 4.2 or later.

Syntax



ntx_wdog [**−d** *Device*]

Description

The **ntx_wdog** command starts the watchdog feature on an adapter board running microcode version 03A or later.

Flags

−d *Device* Specifies the raw device file name of the adapter. Use this option to override the default device name **/dev/rhp0**.

Exit Status

This command returns the following exit values:

0 Successful completion.

>0 An error occurred.

Security

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

Auditing Events: N/A

Examples

To start the watchdog on the adapter **/dev/rhp1**, enter:

```
ntx_wdog −d /dev/rhp1
```

Files

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

/dev/rhp0	Default NTX raw device file name.

Related Information

The **/dev/rhp** file.

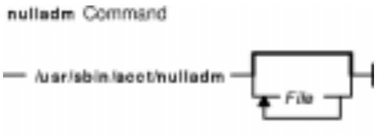
Network Terminal Accelerator Overview in *AIX Versions 3.2 and 4 Asynchronous Communications Guide*.

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 Version 4.3 System Management Guide: Operating System and Devices*.

Setting Up an Accounting System in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

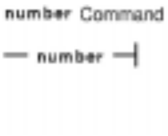
AIX Performance Monitoring and Tuning Commands in *AIX Versions 3.2 and 4 Performance Tuning Guide*.

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.

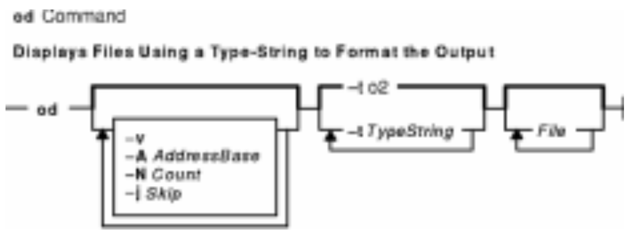
od Command

Purpose

Displays files in a specified format.

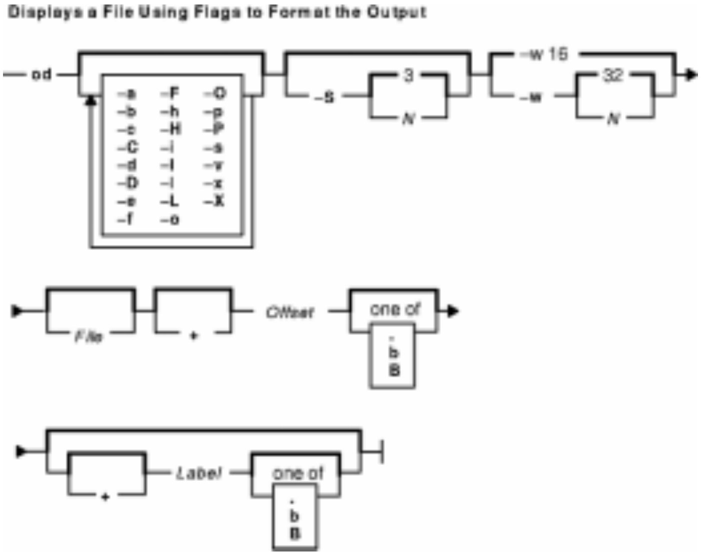
Syntax

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**] [**−I**] [**−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.

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 Version 4.3 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.

−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.

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.

−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. The following nongraphic characters are used as C−language escape sequences:

l

Backslash

l*a*

Alert

l*b*

Backspace

l*f*

Form−feed

l*n*

New−line character

l*0*

Null

l*r*

Carriage return

l*t*

Tab

l*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 **longdouble**, 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:

l

Backslash

l*a*

Alert

l*b*

Backspace

l*f*

Form−feed

l*n*

New−line character

l*0*

Null

l*r*

Carriage return

l*t*

Tab

l*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.

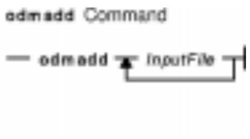
The National Language Support Overview for System Management and Understanding Locale Environment Variables in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

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 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 General Programming Concepts: Writing and Debugging Programs*.

The **odm_add_obj** subroutine.

List of ODM Commands and Subroutines in *AIX General Programming Concepts: Writing and Debugging Programs*.

Understanding ODM Object Classes and Objects in *AIX General Programming Concepts: Writing and Debugging Programs*.

How to Create an Object Class in *AIX General Programming Concepts: Writing and Debugging Programs*.

How to Add Objects to an Object Class in *AIX General Programming Concepts: Writing and Debugging Programs*.

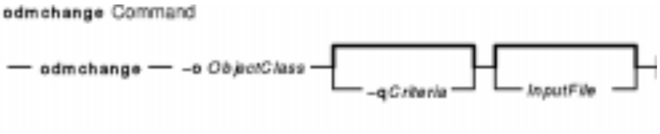
How to Store Object Classes and Objects in *AIX General Programming Concepts: Writing and Debugging Programs*.

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 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 Version 4.3 General Programming Concepts: Writing and Debugging Programs*.

The **odmadd** command.

The **odm_change_obj** subroutine.

Understanding ODM Descriptors in *AIX Version 4.3 General Programming Concepts: Writing and Debugging Programs*.

List of ODM Commands and Subroutines in *AIX General Programming Concepts: Writing and Debugging Programs*.

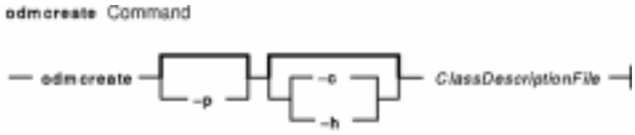
ODM Example Code and Output in *AIX General Programming Concepts: Writing and Debugging Programs*.

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*StdClassNameStdClassNameColNameDescriptorName***;**

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 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 General Programming Concepts: Writing and Debugging Programs*.

The **odm_create_class** subroutine.

List of ODM Commands and Subroutines in *AIX General Programming Concepts: Writing and Debugging Programs*.

Understanding ODM Object Classes and Objects in *AIX General Programming Concepts: Writing and Debugging Programs*.

Understanding ODM Descriptors in *AIX General Programming Concepts: Writing and Debugging Programs*.

How to Create an Object Class in *AIX Version 4.3 General Programming Concepts: Writing and Debugging Programs*.

How to Add Objects to an Object Class in *AIX Version 4.3 General Programming Concepts: Writing and Debugging Programs*.

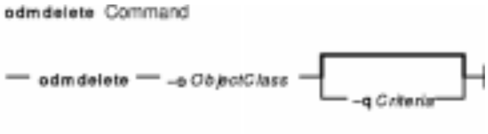
How to Store Object Classes and Objects in *AIX Version 4.3 General Programming Concepts: Writing and Debugging Programs*.

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 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 Version 4.3 General Programming Concepts: Writing and Debugging Programs*.

The **odm_rm_obj** subroutine.

Understanding ODM Object Classes and Objects in *AIX General Programming Concepts: Writing and Debugging Programs*.

List of ODM Commands and Subroutines in *AIX 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 Version 4.3 General Programming Concepts: Writing and Debugging Programs*.

The **odm_rm_class** subroutine.

Understanding ODM Object Classes and Objects in *AIX General Programming Concepts: Writing and Debugging Programs*.

List of ODM Commands and Subroutines in *AIX 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 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 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 Version 4.3 General Programming Concepts: Writing and Debugging Programs*.

Understanding ODM Object Classes and Objects in *AIX General Programming Concepts: Writing and Debugging Programs*.

ODM Example Code and Output in *AIX General Programming Concepts: Writing and Debugging*

Programs.

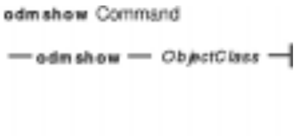
List of ODM Commands and Subroutines in *AIX General Programming Concepts: Writing and Debugging Programs*.

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 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 Version 4.3 General Programming Concepts: Writing and Debugging Programs*.

The **odmcreate** command.

Understanding ODM Object Classes and Objects in *AIX General Programming Concepts: Writing and Debugging Programs*.

ODM Example Code and Output in *AIX General Programming Concepts: Writing and Debugging Programs*.

List of ODM Commands and Subroutines in *AIX General Programming Concepts: Writing and Debugging Programs*.

on Command

Purpose

Executes commands on remote systems.

Syntax

on Command
on
−i
−d
−n
Host
Command
Argument

/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 Version 4.3 System Management Guide: Communications and Networks*.

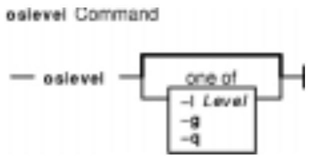
List of NFS Commands.

oslevel Command

Purpose

Reports the latest installed maintenance level of the system.

Syntax



oslevel [**−l***Level* | **−g** | **−q**]

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 filesets at levels earlier than maintenance level specified by the *Level* parameter.

−g Lists filesets at levels later than the current maintenance level.

−q Lists names of known maintenance levels that can be specified using the **−l** flag.

Examples

1. To determine the AIX maintenance level of the system, enter:

   ```
   oslevel
   ```

2. To determine the filesets that are below level 4.1.2.0, enter:

   ```
   oslevel −l 4.1.2.0
   ```

3. To determine the filesets at levels later than the current AIX maintenance level, enter:

   ```
   oslevel −g
   ```

Files

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

Related Information

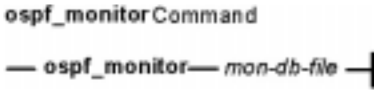
The **lslpp** command.

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_indexremote_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_idtypels_idadv_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

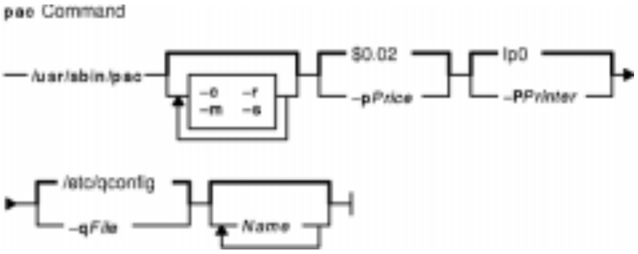
The **gated** Daemon.

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 Version 4.3 System Management Guide: Operating System and Devices*.

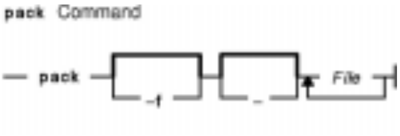
Accounting Overview in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

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 Version 4.3 System User's Guide: Operating System and Devices*.

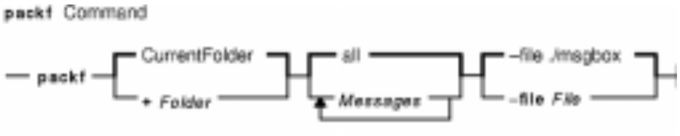
Input and Output Redirection Overview in *AIX Version 4.3 System User's Guide: Operating System and Devices*.

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 Version 4.3 System User's Guide: Communications and Networks*.

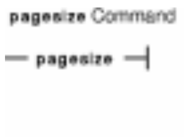
Peek, Jerry. *MH and xmh: E−mail for Users and Programmers.* Sebastopol, CA: O'Reilly & Associates, 1992.

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

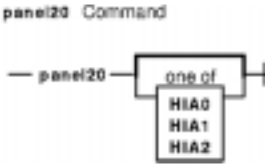
The **getpagesize** subroutine.

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

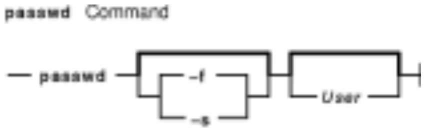
```
panel20 HIA2
```

passwd Command

Purpose

Changes a user's password.

Syntax



passwd [**−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 never appears 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 *.

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.

Security

Access Control: This program should be installed as part of the Trusted Computing Base with execute access for all users. In addition, this program should run **setuid** to the root user in order 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, enter:

   ```
   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, enter:

   ```
   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 enter 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, enter:

   ```
   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 enter 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.

Security Administration in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

Trusted Computing Base Overview in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

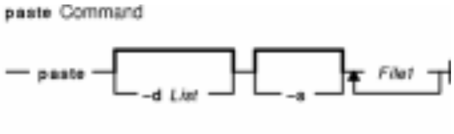
Shells Overview in *AIX Version 4.3 System User's Guide: Operating System and Devices* to learn more about the available shells.

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 2000 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:

l*n*
New−line character

l*t*

Tab

ll

Backslash

l0

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 for Programmers in *AIX General Programming Concepts: Writing and Debugging Programs*.

Files Overview in *AIX Version 4.3 System User's Guide: Operating System and Devices*.

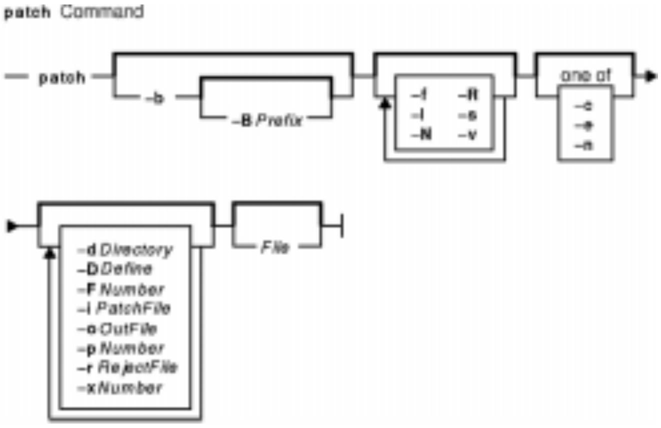
Input and Output Redirection Overview in *AIX Version 4.3 System User's Guide: Operating System and Devices*.

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`:

- **−p0** leaves the entire path name unmodified.
- **−p1** 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

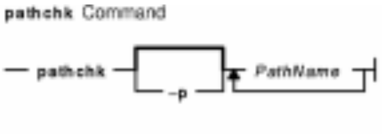
The **diff** command, **ed** command.

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 Overview for System Management in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

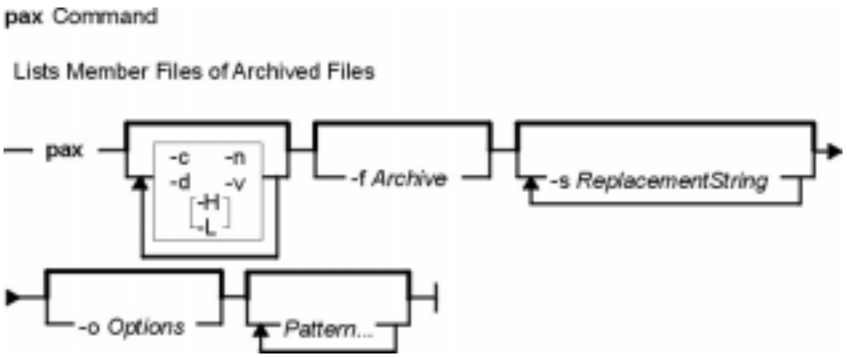
pax Command

Purpose

Extracts, writes, and lists members of archive files; copies files and directory hierarchies.

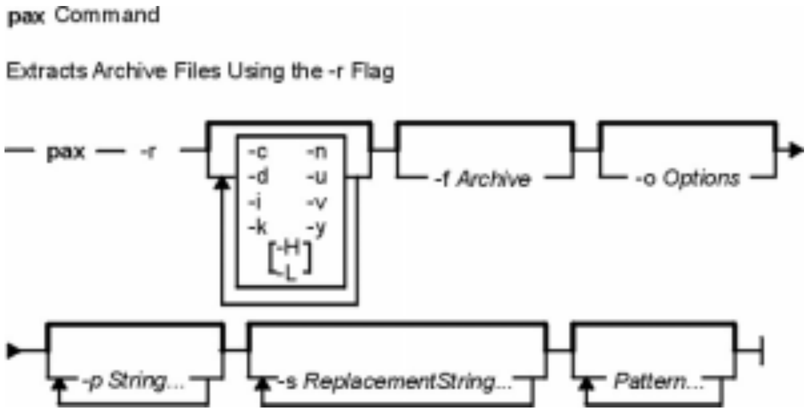
Syntax

To List Member Files of Archived Files



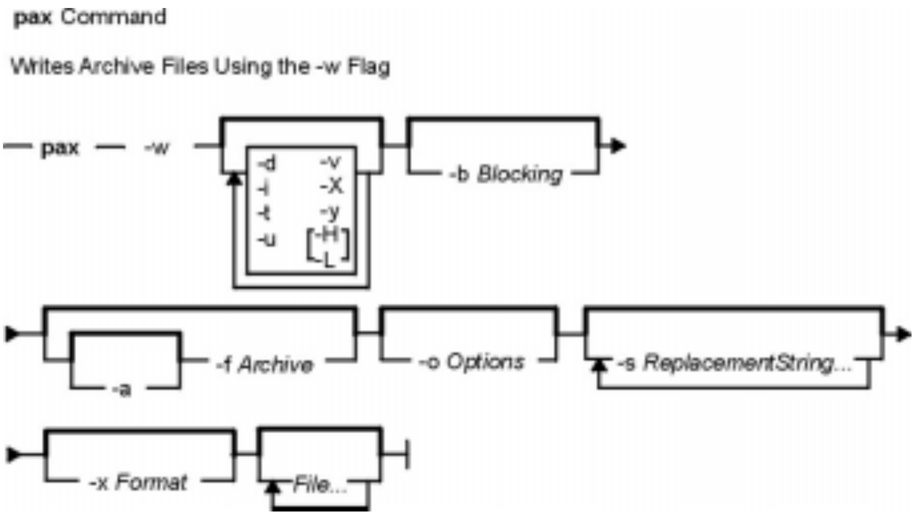
pax [**−c**] [**−d**] [**−n**] [**−v**] [**−H** | **−L**] [**−f** *Archive*] [**−s** *ReplacementString* ...]
[**−o** *Options*] [*Pattern* ...]

To Extract Archive Files Using the −r Flag



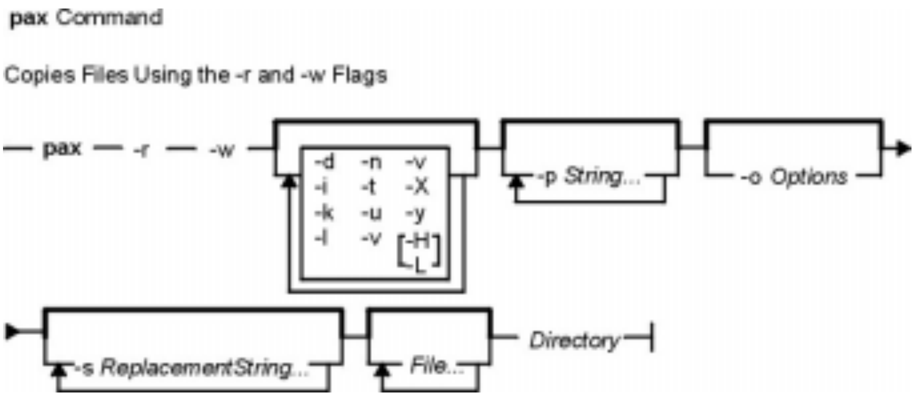
pax −r [**−c**] [**−d**] [**−i**] [**−k**] [**−n**] [**−u**] [**−v**] [**−H** | **−L**] [**−y**] [**−f** *Archive*]
[**−o** *Options*] [**−p** *String* ...] [**−s** *ReplacementString* ...] [*Pattern* ...]

To Write Archive Files Using the −w Flag



pax −w [**−d**] [**−i**] [**−t**] [**−u**] [**−v**] [**−X**] [**−H** | **−L**] [**−y**] [**−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**] [**−y**] [**−p** *String* ...] [**−o** *Options*] [**−s** *ReplacementString* ...] [*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**, **−v** and **−y** 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**, **−X**, and **−y** 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**, **−X**, and **−y** 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. 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.*"` would suppress security−related information.

exthdr.name=*string* (Applicable only to the **−x pax** format.)

This keyword alows 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.

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.

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 unencoded binary data.

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 pre−existing 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.
- 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 32768 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 32768 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 32768 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.

−y Prompts the user for the disposition of each file or archive member. For each member that matches the *Pattern* parameter or file that matches the *File* parameter, a prompt is displayed that contains the name of the file or archive member. If an affirmative response is entered, then the file is processed; otherwise, the file is ignored.

Flag Interaction and Processing Order

The flags that operate on the names of files or archive members (**−c**, **−i**, **−n**, **−s**, **−u**, **−v**, and **−y**) 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**, **−y**, 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**, **−y**, and **−i** flags modify, in that order, the names resulting from these modifications.
- 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.

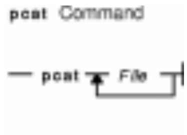
The Files Overview in *AIX Version 4.3 System User's Guide: Operating System and Devices*.

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.

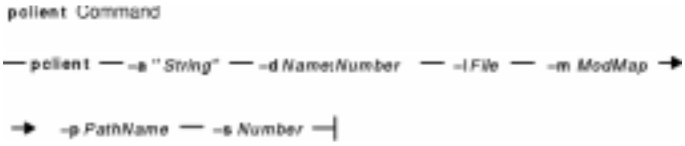
Files Overview.and Input and Output Redirection Overview in *AIX Version 4.3 System User's Guide: Operating System and Devices*.

pclient Command

Purpose

Opens the connection to an Xstation for a specified period of time.

Syntax



pclient−a "*String***" −d***Name*:*Number* **−l***File* **−m***ModMap* **−p***PathName* **−s***Number*

Description

The **pclient** command holds the connection to an Xstation open for the amount of time specified by the **−s** flag allowing commands to be sent and received.

Flags

−a "*String***"**	Specifies the command that initializes the Xwindows environment. *String* consists of the **aixterm** command and selected flags, and must be enclosed in quotation marks. The string is executed using the **ksh** shell. Other commands, such as the **xset** command, may be included and are separated with ; (semicolon).
−d*Name*: *0*	Specifies the name and number of the display.
−l*File*	Specifies the name of the keyboard map file.
−m*ModMap*	Specifies the full name of the keyboard xmodmap to be executed.
−p*PathName*	Specifies the path to the keyboard map file.
−s*Number*	Specifies the time in seconds that the connection is held open.

Security

Access Control: Only the root user should have execute (x) access to this command.

Examples

The following stanza from the **/etc/x_st_mgr/x_st_mgrd.cf** file shows the normal use of the command for an Xstation named `taylor`:

```
taylor DISPLAY=taylor:0; export DISPLAY;\
       XSTATION=taylor; export XSTATION;\
       LANG=En_US; export LANG;\
       /usr/lpp/x_st_mgr/bin/pclient\
       −p /usr/lpp/X11/defaults/xmodmap/En_US\
      −l keyboard \
       −m /usr/lpp/X11/bin/xmodmap \
       −s 5 \
     −a "/usr/lpp/X11/bin/aixterm\
          −fn Rom14 −geometry 80x25+0+0 \
          −W −e /usr/lpp/x_st_mgr/bin/login"
```

The example includes flags inside the **−a"***String***"** that specify the font (`−fn`), the location and dimensions of the window (`−geometry`), the location of the mouse cursor (`−W`), and the login command to be executed in the window (`−e`).

Files

/usr/lpp/x_st_mgr/bin/pclient	Contains the **pclient** command.
/usr/lpp/X11/bin/aixterm	Contains the **aixterm** command.

Related Information

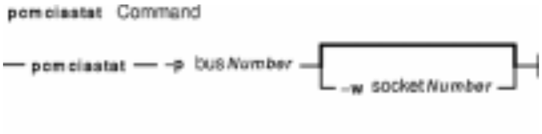
The **aixterm** command.

pcmciastat Command

Purpose

Displays the current sockets' status and defined PCMCIA devices.

Syntax



pcmciastat −p bus*Number* [**−w** socket*Number*]

Description

The **pcmciastat** command displays the current sockets' status and defined PCMCIA devices in the Object Data Manager (ODM) database.

You may insert or remove the PCMCIA card while the system is running and it will automatically configure itself. You can also use the Web−based System Manager Devices application (**wsm devices** fast path) to configure or unconfigure the card or use SMIT or the **mkdev** or **rmdev** command.

Flags

−p bus*Number* Specifies the PCMCIA BUS device in configured state in ODM identified by bus*Number*. This flag is required.

−w socket*Number* Specifies the socket to display status for.

Security

Access Control: Any User

Auditing Events: N/A

Example

To display the current sockets' status and defined PCMCIA devices for PCMCIA BUS bus2, enter:

```
pcmciastat −p bus2
```

Files

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

Related Information

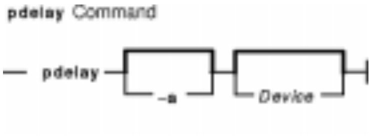
Setting up and running Web−based System Manager in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

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

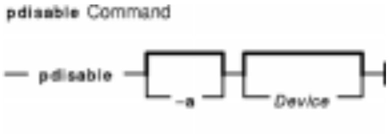
The **getty** command, **init** command, **pdisable** command, **penable** command, **phold** command, **pshare** command, **pstart** command.

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.

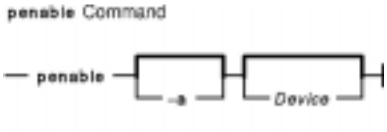
The **inittab** file.

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.

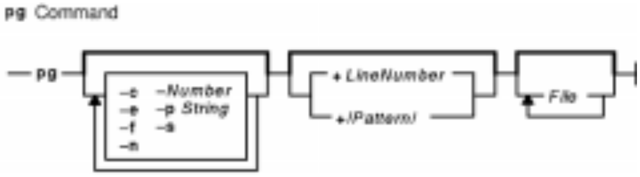
The **inittab** file.

pg Command

Purpose

Formats files to the display.

Syntax



pg [**−***Number*] [**−c**] [**−e**] [**−f**] [**−n**] [**−p** *String*] [**−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.

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.

−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 Version 4.3 System Management Guide: Operating System and Devices*.

Input and Output Redirection Overview and Shells Overview in the *AIX Version 4.3 System User's Guide: Operating System and Devices*.

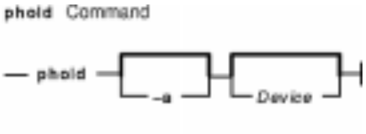
Files Overview in the *AIX Version 4.3 System User's Guide: Operating System and Devices*.

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.

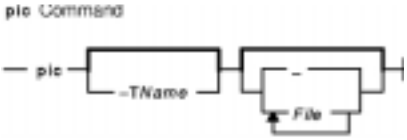
The **inittab** file.

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, in inches, of the picture, 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 *XCommandlineX*

[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 # (comment 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.

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*
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 normally 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***NthShape*

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 troff, 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
|	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, **troff** command.

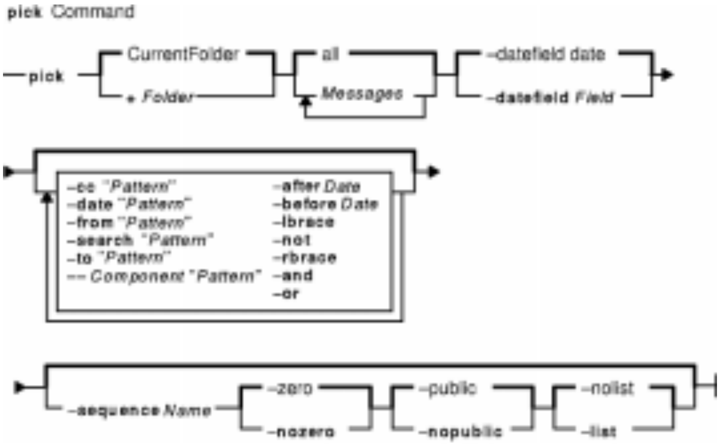
The **.PE** macro, **.PF** macro, **.PS** macro, **me** macro package.

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 Version 4.3 System User's Guide: Communications and Networks*.

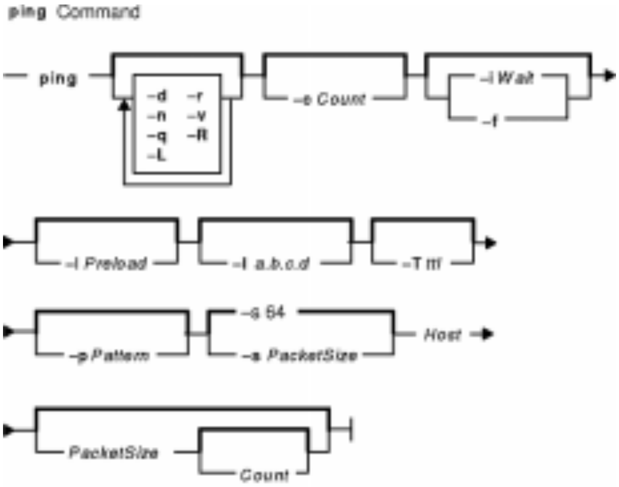
Peek, Jerry. *MH and xmh: E−mail for Users and Programmers.* Sebastopol, CA: O'Reilly & Associates, 1992.

ping Command

Purpose

Sends an echo request to a network host.

Syntax



ping [**−d**] [**−n**] [**−q**] [**−r**] [**−v**] [**−R**] [**−c** *Count*] [**−f** | **−i** *Wait*] [**−l** *Preload*] [**−p** *Patte*
[**−s** *PacketSize*] [**−L**] [**− I***a.b.c.d.*] [**−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).

−d Starts socket−level debugging.

−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 multicasts. The **−I** flag is an uppercase i.

−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.

−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.

−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.

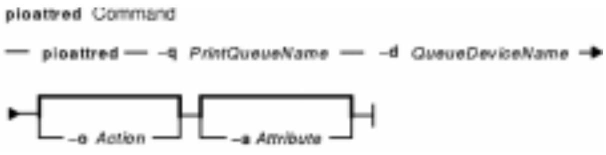
Network Overview in *AIX Version 4.3 System Management Guide: Communications and Networks*.

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.

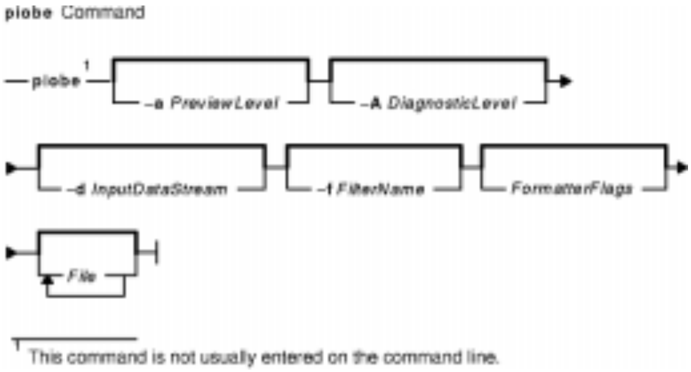
Virtual Printer Definitions and Attribute Values in the *AIX Version 4.3 Guide to Printers and Printing*.

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 *InputDataTyp*e 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 can be useful for diagnosing errors encountered by a pipeline of filters that is processing a print file, a header page, or a trailer page. 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.

A value of **1** is recommended. A value of **0** can be 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 **3** can be 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 Version 4.3 Guide to Printers and Printing*.

Understanding the Interaction between qdaemon and the Backend in the *AIX Version 4.3 Guide to Printers and Printing*.

Understanding Backend Routines in libqb in the *AIX Version 4.3 Guide to Printers and Printing*.

Printer Addition Management Subsystem: Programming Overview in *AIX Kernel Extensions and Device Support Programming Concepts*.

Configuring a Printer without Adding a Queue in the *AIX Version 4.3 Guide to Printers and Printing*.

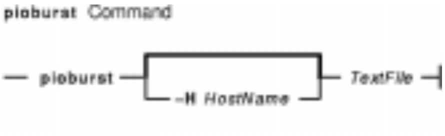
Adding a Printer Using the Printer Colon File in the *AIX Version 4.3 Guide to Printers and Printing*.

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 Version 4.3 Guide to Printers and Printing*.

Printer Colon File Conventions in the *AIX Version 4.3 Guide to Printers and Printing*.

Printer Colon File Escape Sequences in the *AIX Version 4.3 Guide to Printers and Printing*.

Printer Code Page Translation Tables in the *AIX Version 4.3 Guide to Printers and Printing*.

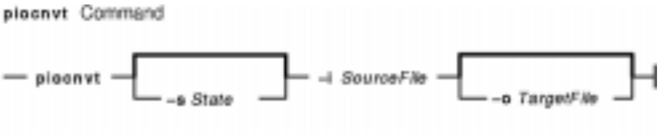
Virtual Printer Definitions and Attributes in the *AIX Version 4.3 Guide to Printers and Printing*.

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 Version 4.3 Guide to Printers and Printing*

Spooler Overview for System Management in the *AIX Version 4.3 Guide to Printers and Printing*.

Printer Specific Information in the *AIX Version 4.3 Guide to Printers and Printing*.

Printer Support in the *AIX Version 4.3 Guide to Printers and Printing*.

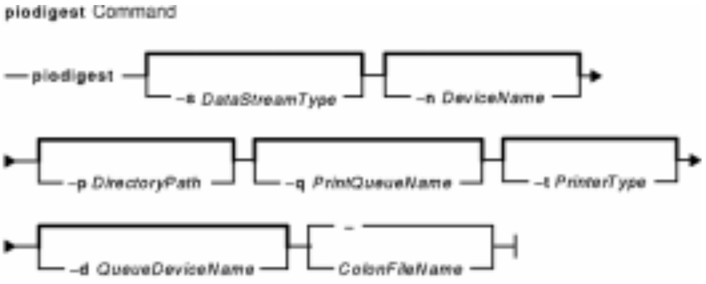
Virtual Printer Definitions and Attributes in the *AIX Version 4.3 Guide to Printers and Printing*.

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.

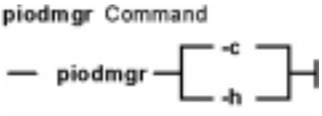
The Printer Overview for System Management and Virtual Printer Definitions and Attributes, Printer Code Page Translation Tables, Printer Colon File Conventions, Printer Colon File Escape Sequences, Printer Backend Overview for Programming, Understanding the Interaction between qdaemon and the Backend, Understanding Backend Routines in libqb in *AIX Version 4.3 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 Version 4.3 Guide to Printers and Printing*.

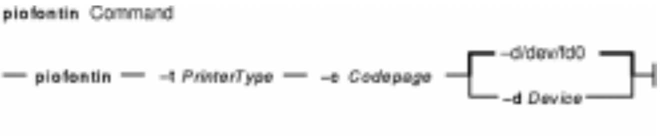
Object Data Manager (ODM) Overview for Programmers in *AIX Version 4.3 General Programming Concepts: Writing and Debugging Programs*.

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 Version 4.3 Guide to Printers and Printing*.

pioformat Command

Purpose

Drives a printer formatter.

Syntax

pioformat Command
pioformat
-@ DataBaseFile
-! FormatterName
-# + PassThroughOption

/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 Version 4.3 Guide to Printers and Printing*.

Virtual Printer Definitions and Attributes in the *AIX Version 4.3 Guide to Printers and Printing*.

Printer Addition Management Subsystem: Programming Overview in *AIX Kernel Extensions and Device Support Programming Concepts*.

Printer Code Page Translation Tables in the *AIX Version 4.3 Guide to Printers and Printing*.

Printer Colon File Conventions in the *AIX Version 4.3 Guide to Printers and Printing*.

Printer Colon File Escape Sequences in the *AIX Version 4.3 Guide to Printers and Printing*.

Printer Specific Information in the *AIX Version 4.3 Guide to Printers and Printing*.

Printer Support in the *AIX Version 4.3 Guide to Printers and Printing*.

Printer Backend Overview for Programming in the *AIX Version 4.3 Guide to Printers and Printing*.

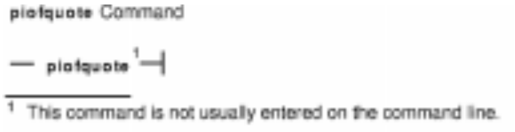
Adding a Printer Using the Printer Colon File in the *AIX Version 4.3 Guide to Printers and Printing*.

piofquote Command

Purpose

Converts certain control characters destined for PostScript printers.

Syntax



1 This command is not usually entered on the command line.

/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.

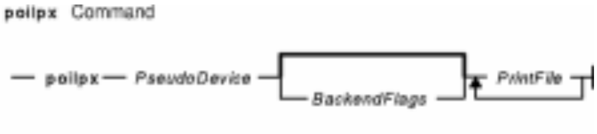
Printer Overview for System Management in the *AIX Version 4.3 Guide to Printers and Printing*.

piolpx Command

Purpose

Provides printer backend support for Xstation attached printers.

Syntax



piolpx*PseudoDevice* [*BackendFlags*] *PrintFile ...*

Description

The **piolpx** command accepts the name of files to be printed and printer backend flags and their arguments, reads the pseudo device file to determine the communication characteristics, and then invokes the printer backend **piobe** command. The output from the **piolpx** command is piped to the **catlpx** program.

The command builds flags for the **catlpx** program in the following communications characteristics format:

```
catlpx XStationName               /* For parallel   */
catlpx XStationName −s[0−4]Baud,Parity,Bits,Stops
                                /* For serial     */
```

Finally, the **piolpx** command passes print file names and backend flags, if any are specified, to the backend in the following format:

```
piobe [BackendFlags] PrintFile ... | catlpx ...
```

Flags

BackendFlags Specifies the backend flags listed in the **qprt** command. The **piolpx** command supports all the **qprt** command backend flags.

bpc Designates the bits per character for the pseudo device.

parity Indicates parity within the pseudo device with the following values:

Symbol	**Meaning**
'n'	None
'o'	Odd
'e'	Even
'so'	Stuck odd

'se'

Stuck even

port Designates the port. Values for the port number are **p** for parallel, **s** for a single serial port, or **s[***0−4*] when there is more than one serial port.

PrintFile Indicates the name of the file or files to be printed to the virtual printer.

PseudoDevice Indicates the pseudo device file to be read. The **piolpx** command reads the pseudo device file to determine the communication characteristics of the device as follows:

speed Designates the baud rate in the pseudo device.

stops Designates the number of stop bits in the pseudo device.

Example

To print the `NewFile` file with a pitch value of `12` to the Xstation named `Einstein` with a printer attached to the parallel port, enter

```
/usr/lib/lpd/pio/etc/piolpx p@Einstein −o −p −o 12 NewFile
```

Files

/usr/lib/lpd/pio/etc/piolpx	Contains the **piolpx** command.
/usr/lib/lpd/piobe	Contains the printer backend.
/usr/lpp/x_st_mgr/bin/catlpx	Contains the Xstation print command.
/usr/lib/lpd/pio/etc/piomgpdev	Contains the pseudo device manager.

Related Information

The **piobe** command, **qprt** command.

Printer Overview for System Management in the *AIX Version 4.3 Guide to Printers and Printing*.

Spooler Overview in the *AIX Version 4.3 Guide to Printers and Printing*.

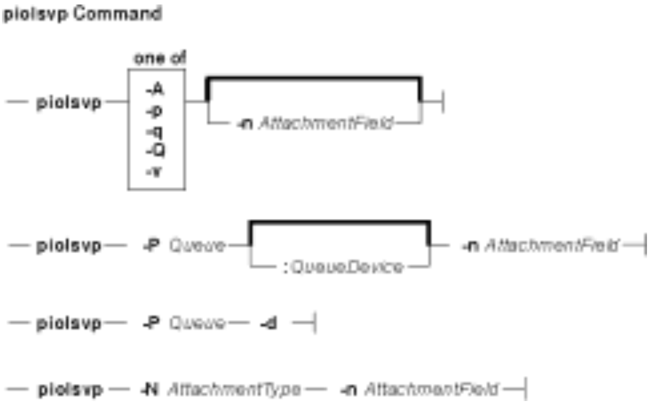
Printer Backend Overview for Programming in the *AIX Version 4.3 Guide to Printers and Printing*.

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. For example, the entry for an Xstation attachment type might be **xsta.attach**.

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
   xsta                    Xstation 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 Version 4.3 Guide to Printers and Printing*.

Printer Overview for System Management in the *AIX Version 4.3 Guide to Printers and Printing*.

Spooler Overview for System Management in the *AIX Version 4.3 Guide to Printers and Printing*.

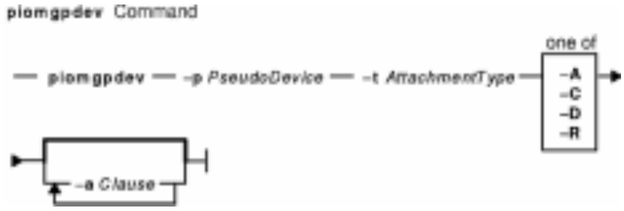
Printer Backend Overview for Programming in the *AIX Version 4.3 Guide to Printers and Printing*.

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. For example, the information about an Xstation named `yoder` and a port called `cs2` might be:

```
Xstation = yoder
port = s2
baud = 9600
stop_bits = 1
```

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.

Examples

1. To add a pseudo−device for an Xstation attachment for the Xstation named `yoder` and the parallel port named `p`, enter on one line:

   ```
   piomgpdev −p p@yoder −A −a 'Xstation=yoder' −a
   'interface=parallel' −a 'port=p'
   ```

2. To change a pseudo−device for an Xstation attachment for the Xstation `yoder` and the serial port `s2`, enter:

   ```
   piomgpdev −p s2@yoder −C −a 'baud=9600'
   ```

3. To remove a pseudo−device for an Xstation attachment for the Xstation `yoder` and the parallel port `p`, enter:

   ```
   piomgpdev −p p@yoder −R
   ```

4. To display information regarding the baud rate for an Xstation attachment for the Xstation `yoder` and the parallel port `p`, enter:

   ```
   piomgpdev −p p@yoder −D −a baud
   ```

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 Version 4.3 Guide to Printers and Printing*.

Spooler Overview for System Management in the *AIX Version 4.3 Guide to Printers and Printing*.

Printer Backend Overview for Programming in the *AIX Version 4.3 Guide to Printers and Printing*.

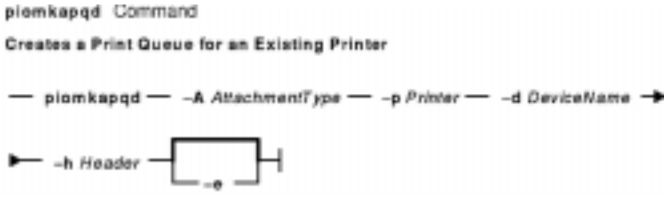
piomkapqd Command

Purpose

Builds a SMIT dialog to create print queues and printers.

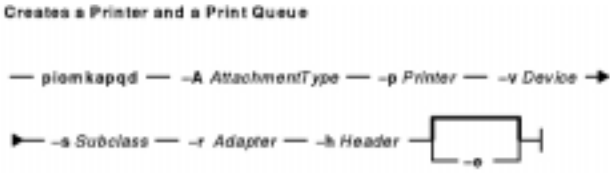
Syntax

To Create a Print Queue for an Existing Printer



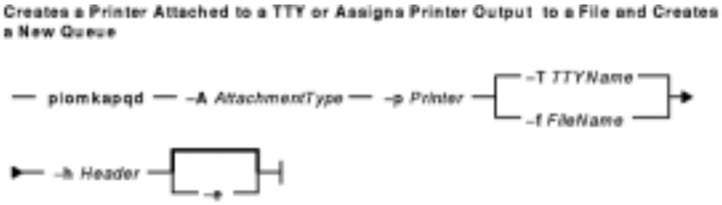
piomkapqd −A *AttachmentType***−p** *Printer***−d** *DeviceName* **−h** *Header* [**−e**]

To Create a Printer and a Print Queue



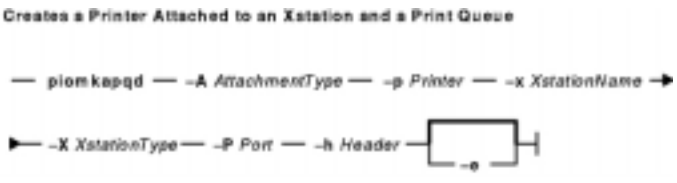
piomkapqd −A *AttachmentType***−p** *Printer***−v** *Device* **−s** *Subclass* **−r** *Adapter***−h** *Header* [**−e**]

To Create a Printer Attached to a TTY or to Assign Printer Output to a File and Create a New Queue


Creates a Printer Attached to a TTY or Assigns Printer Output to a File and Creates a New Queue
piomkapqd
-A AttachmentType
-p Printer
-T TTYName
-f FileName
-h Header
-e

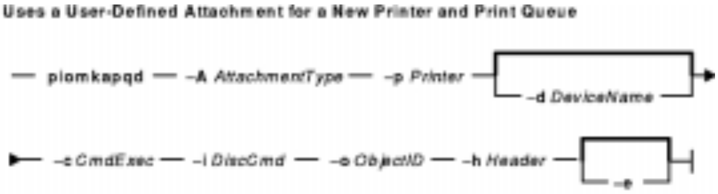
piomkapqd −A *AttachmentType***−p** *Printer* { **−T** *TTYName* | **−f** *FileName* } **−h** *Header* [**−e**]

To Create a Printer Attached to an Xstation and a Print Queue



piomkapqd −A *AttachmentType***−p** *Printer***−x** *XstationName* **−X** *XstationType* **−P** *Port*
−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 Mangement Infterface 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.

xsta
Specifies an Xstation 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`.

−P *Port* Specifies the Xstation port number where the printer is attached.

−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.

−x*XstationName* Specifies the name of the Xstation.

−X*XstationType* Specifies the type of the Xstation.

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 new printer attached to an Xstation named `Aegean` and creates a new print queue, enter:

   ```
   piomkapqd −A xsta −p ibm4029 −x Aegean −X 120 −P s1 −h 'Add Xstation Printer'
   ```

4. 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'
   ```

5. 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
   ```

6. 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 Version 4.3 Guide to Printers and Printing*.

System Management Interface Tool (SMIT) Overview in the *AIX Version 4.3 System Management Guide: Operating System and Devices*.

Printer Backend Overview for Programming in the *AIX Version 4.3 Guide to Printers and Printing*.

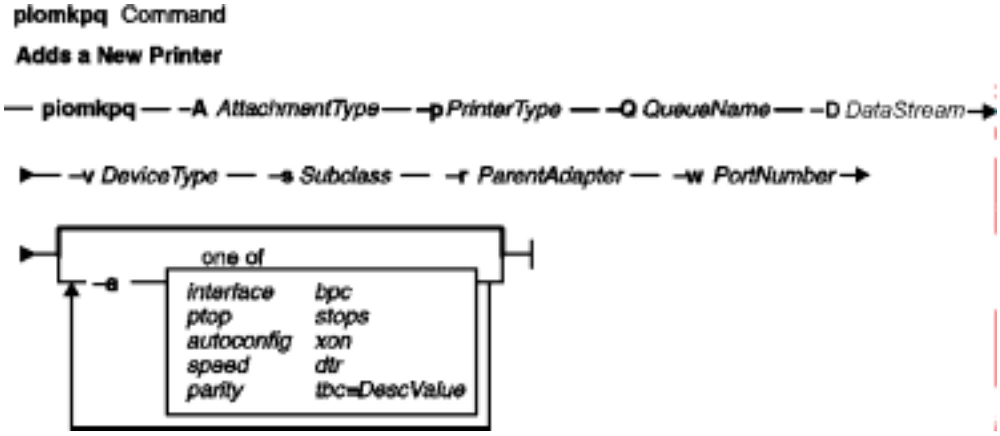
piomkpq Command

Purpose

Creates a print queue.

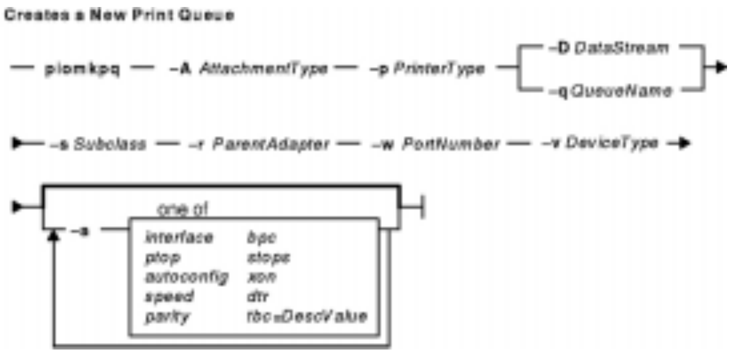
Syntax

To add a new printer



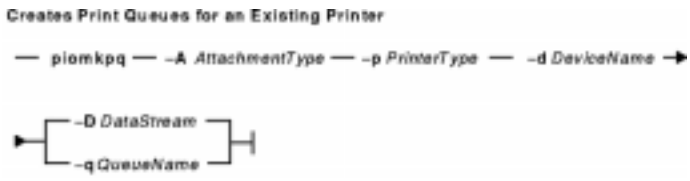
piomkpq −A *AttachmentType*−**p**   *PrinterType* **−Q** *QueueName* **−D** *DataStream* **−v** *DeviceType*
−s *Subclass*−**r** *ParentAdapter*−**w** *PortNumber* [**−a** { *interface* | *ptop* | *autoconfig* | *speed* | *parity* | *bpc* |
...

To create a new print queue



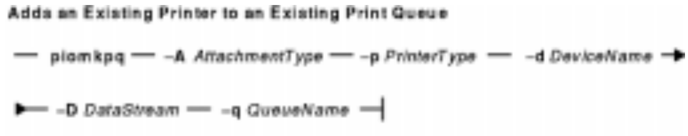
piomkpq −A *AttachmentType*−**p**   *PrinterType* { **−D** *DataStream* | **−q** *QueueName* }
−s *Subclass*−**r** *ParentAdapter*−**w** *PortNumber*−**v** *DeviceType* [**−a** { *interface* | *ptop* | *autoconfig* | *speed* | *par*
...

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.
xsta
Specifies an Xstation 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 add a new printer to an existing Xstation print queue named bob, enter:

```
piomkpq −A xsta −p ibm4019 −Q bob −D pcl −x Einstein −t 120 −P s1
[−a speed=19200]
```

5. 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 Version 4.3 Guide to Printers and Printing* .

Spooler Overview for System Management in *AIX Version 4.3 Guide to Printers and Printing* .

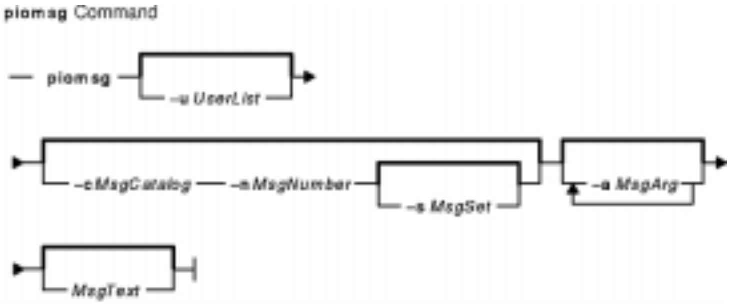
Printer Backend Overview for Programming in *AIX Version 4.3 Guide to Printers and Printing* .

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$sprintf** 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$sprintf** 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 Version 4.3 Guide to Printers and Printing* .

Spooler Overview for System Management in *AIX Version 4.3 Guide to Printers and Printing* .

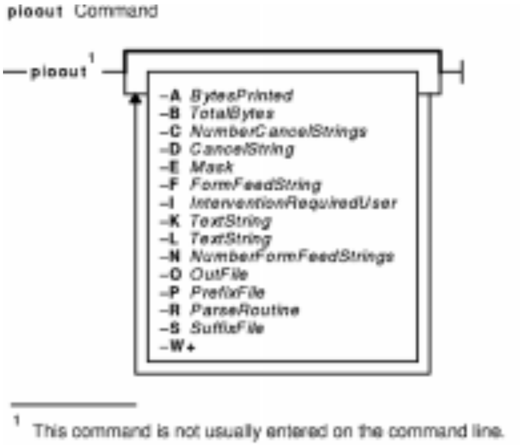
Printer Backend Overview for Programming in *AIX Version 4.3 Guide to Printers and Printing* .

piooout Command

Purpose

Printer backend's device driver interface program.

Syntax



/usr/lpd/pio/etc/piooout [**−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 **piooout** 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 **piooout** 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 General Programming Concepts: Writing and Debugging Programs*.

Printer Addition Management Subsystem: Programming Overview in *AIX Kernel Extensions and Device Support Programming Concepts*.

Printer Code Page Translation Tables in *AIX General Programming Concepts: Writing and Debugging Programs*.

Printer Colon File Conventions in *AIX General Programming Concepts: Writing and Debugging Programs*.

Printer Colon File Escape Sequences in *AIX General Programming Concepts: Writing and Debugging Programs*.

Understanding the Interaction between qdaemon and the Backend in *AIX General Programming Concepts: Writing and Debugging Programs*.

Understanding Backend Routines in libqb in *AIX General Programming Concepts: Writing and Debugging Programs*.

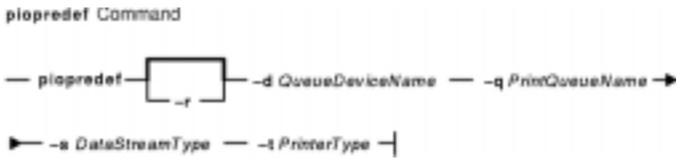
How to Add a Printer Using the Printer Colon File in *AIX General Programming Concepts: Writing and Debugging Programs*.

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 Version 4.3 Guide to Printers and Printing* .

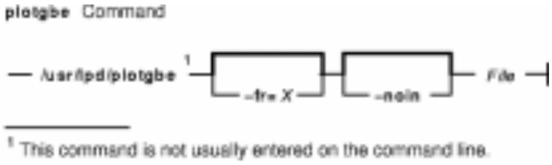
Printer Addition Management Subsystem: Programming Overview in *AIX Kernel Extensions and Device Support Programming Concepts*.

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 Version 4.3 Guide to Printers and Printing* .

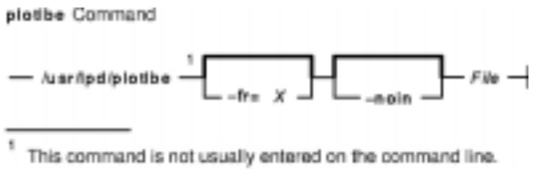
Adding Plotter Support with 5080 in *AIX Version 4.3 Guide to Printers and Printing* .

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/plotlbe Contains the **plotlbe** command.

Related Information

The **enq** command, **plotgbe** command.

Printer Overview for System Management in *AIX Version 4.3 Guide to Printers and Printing* .

pmctrl Command

Purpose

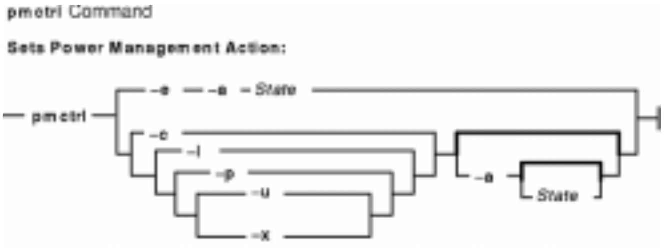
Controls or queries Power Management information.

Syntax

To Set Power Management Device:

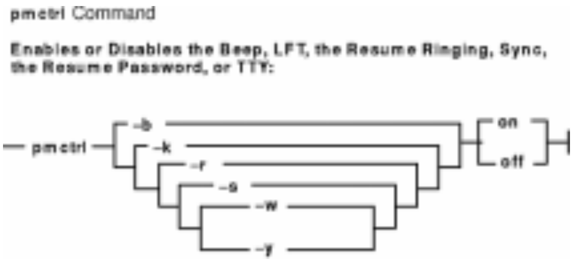
pmctrl −d [[*DeviceName* [**−t** [*IdleTime StandbyTime*]] |
[**lft0** [**−t** [*DimTime SuspendTime TurnOffTime*]]]]

To Set Power Management State:



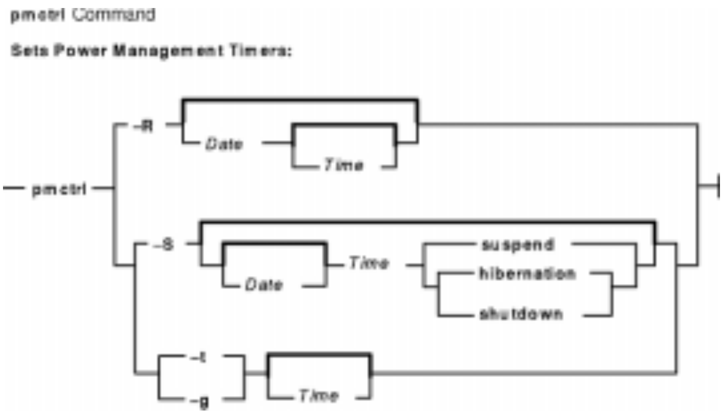
pmctrl { **−e** **−a** *State* } | { **−c** | **−l** | **−p** | **−u** | **−x** } [**−a** *State*]

To Enable or Disable the Beep, LFT, the Resume Ringing, Sync, the Resume Password, or TTY:


pmctrl Command
Enables or Disables the Beep, LFT, the Resume Ringing, Sync, the Resume Password, or TTY:
pmctrl
−b
−k
−r
−s
−w
−y
on
off

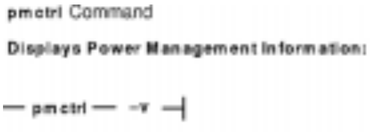
pmctrl { **−b** | **−k** | **−r** | **−s** | **−w** | **−y** | } [**on** | **off**]

To Set Power Management Timers:



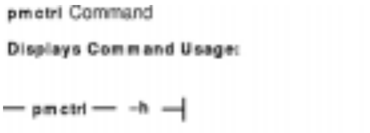
pmctrl { **−R** [[*Date*] *Time*] |
−S [[*Date*] *Time* **suspend** | **hibernation** | **shutdown**]] | **−t** [*Time*] | **−g** [*Time*] }

To Display Power Management Information:



pmctrl −v

To Display Command Usage:



pmctrl−h

Description

The **pmctrl** command controls or queries Power Management information. The **pmctrl** command lets you control the *State* of the system when any of the following circumstances occur provided you have the necessary support allowing you to control the state:

- The lid of the notebook is closed (Lid close)
- The battery is low (Low battery)
- The mainpower switch is pushed (Main power switch)
- The system idle timer has run out (System idle timer expiration)

The *State* of the system can be any of the following:

- **full_on**
- **enable**
- **standby**
- **suspend**
- **hibernation**
- **shutdown**
- **ignore** (not valid with the **−e** flag)

Flags

−a [*State*]

Displays the current system state when used alone.

When used with the parameter *State* and in combination with other flags, specifies the system state to change to.

−b [**on** | **off**]

Displays the current system setting for beep when used alone.

When used with the parameter **on** or **off**, enables or disables the beep whenever the system changes state.

−c

Displays the current system setting for Lid close.

When used with the **−a***State* flag, sets the system state to change to (transition) whenever you close the lid of the notebook.

−d*DeviceName*

Displays the list of all Power Management aware Device Drivers when used alone.

When used with *DeviceName* or *DeviceName***−t**, displays the idle time and standby time of the device specified by *DeviceName*.

When used with *DeviceName***−t***IdleTimeStandbyTime* , where *DeviceName* is NOT **lft0**, sets the idle time and standby time of the device specified by *DeviceName*.

> **Note:** The *DeviceName*, **lft0**, specifies the Display Power Management System (DPMS).

When used with **lft0−t***DimTimeSuspendTimeTurnoffTime*, sets the dim time, suspend time, and turnoff time of **lft0**.

The minimum value for all the *Time* arguments is 1 minute, and the maximum value is 120 minutes. Specifying −1 as a *Time* argument means no change. Specifying 0 (zero) in any *Time* argument disables the timer.

−e

Requests a transition to the specifeid state. This flag must be used with the **−a***State* flag.

−h

Displays the long command usage.

−g [*Time*]

Displays the current system setting for the duration of the suspend to hibernation when used alone.

When used with the *Time* parameter and the **−a***State* flag, sets duration of suspend to hibernation in minutes. Once this is set, after the system has changed its state to suspend, this timer begins to tick. When the timer expires, the system goes to hibernation state. The minimum value of *Time* is 1 minute, and the maximum value is 120 minutes. Specifying 0 (zero) disables the timer.

−k [**on** | **off**]

Displays the current system setting for Low Function Terminal (LFT) termination when used alone.

When set to **on**, the lft session will be terminated when the system enters suspend or hibernation state. When set to **off**, the lft session will not be terminated when the system enters suspend or hibernation state.

−l

(lowercase L) Displays the current system setting for Low battery.

When used with the **−a***State* flag, sets the system state to change to (transition) whenever the optional battery is low.

−p

Displays the current system setting for Main power switch.

When used with the **−a***State* flag, sets the system state to change to (transition) whenever the main power switch is pressed.

> **Note:** If necessary, the selected state transition can be overridden and the power immediately turned off by pressing the power button three times quickly (within a two second period).

−r [**on** | **off**]

Displays the current system setting for ringing resume when used alone. This setting is used when connected to a telephone through a modem.

When set to **on**, the ring of the telephone will cause the system to resume from hibernation. When set to **off**, the ring of the telephone will not cause the system to resume from suspend or hibernation.

−R [[*Date*] *Time*]

Displays the current system setting of the timer for system resume from a power−saving state (**hibernation** on a workstation or **suspend** on a notebook computer) when used alone.

When used with the parameters, [*Date*] *Time*, sets the alarm timer for system resume. Specify [*Date*] *Time*, with the format [YYMMDD]hhmm. Specifying only *Time* means everyday. Setting *Time* to 0 (zero) means disable the timer. Setting *Time* to 0000 means midnight.

−s [**on** | **off**]

Displays the current system setting for sync daemon termination when used alone.

> **Note:** If this flag is on, the sync daemon will be terminated when entering standby state, but reinvoked when resumed from standby.

When set to **on**, the sync daemon will be terminated when the system enters standby state. When set to **off**, the sync daemon will not be terminated when the system enters standby state.

−S [[*Date*] *Time*] **suspend** | **hibernation** | **shutdown**

Displays the current system setting of the timer for system to enter a power−saving state (**suspend**, **hibernation** or **shutdown**) when used alone.

When used with the parameters, [*Date*] *Time*, sets the alarm timer for entering a power−saving state. Specify [*Date*] *Time*, with the format [YYMMDD]hhmm. Specifying only *Time* means everyday. Setting *Time* to 0 (zero) means disable the timer. Setting *Time* to 0000 means midnight.

−t [*Time*]
Displays the current system setting of the system idle−time when used alone.

When used with the *Time* parameter, sets the system idle−time in minutes. Once this is set, this timer begins to tick while the system is left idle. When the time expires, the system changes to the state specified by the **pmctrl −xa** command. The minimum value of *Time* is 1 minute, and the maximum value is 120 minutes. Specifying 0 (zero) disables the timer.

−u
Displays the current system setting of the maximun state (**shutdown** > **hibernation** > **suspend** > **standby** > **ignore**) permitted to general users. You must have root authority to set this flag. This flag does not cause any system state change.

When used with the **−a***State* flag, sets the maximum system state that a general user can set. For example, if a user with root authority enters:

```
pmctrl −u −a suspend
```

then a general user will not be able to invoke **pmctrl −e −a hibernation** or **pmctrl −e −a shutdown**, but would be able to invoke **pmctrl −e −a suspend** or **pmctrl −e −a standby**. because the maximum state permitted to the general user is **suspend**.

If **pmctrl −ua ignore** is entered, general users can not request any state transition.

−w [**on** | **off**]
Displays the current system setting, when the X server is running,for password query resume when used alone.

When used with the parameter **on** or **off**, enables or disables the password query.

−v
Displays all Power Management information.

−x
Displays the current system setting of the system transistion for when the system idle−timer expires.

When used with the **−a***State* flag, sets the system transistion state when the system idle−timer expires

−y [**on** | **off**]
Displays the current system setting for TTY termination when used alone.

When set to **on**, the TTY session will be terminated when the system enters suspend or hibernation state. When set to **off**, the TTY session will not be terminated when the system enters suspend or hibernation state.

Security

Access Control: Any User

Auditing Events: N/A

Examples

1. To display all Power Management Information, enter:

   ```
   pmctrl −v
   ```

   Something similar to the following displays:

   ```
   Current system state is enable.
   List of registered PM aware DDS is as follows:
   lft0 (DPMS)
   mouse0
   sa0
   kbd0
   cd0
   ent1
   ppa0
   fda0
   bus2
   hdisk0
   scsi0
   System idle time is set to 20. (in minute)
   Duration of suspend to hibernation is set to 20. (in minute)
   Action for notebook lid close is transition to suspend.
   Action for main power switch off is transision to hibernation.
   Action for low battery is suspend.
   Action for system idle timer expiration is transition standby.
   Maximum state authorized for general users is set to hibernation.
   Beep setting is on.
   Resume password setting is set to on.
   Ringing resume setting is set to off
   LFT termination setting is set to off.
   TTY termination setting is set to off.
   Syncd termination setting is set to off.
   Timer for suspend is set to disabled.
   Timer for resume is set to disabled.
   ```

2. To display the current system state, enter:

   ```
   pmctrl −a
   ```

   Something similar to the following displays:

   ```
   Current system state is enable.
   ```

3. To immediately start a state transition to suspend state, enter:

   ```
   pmctrl −e −a suspend
   ```

4. To start a state transition to hibernation whenever the lid is closed, enter:

   ```
   pmctrl −c −a hibernation
   ```

5. To start a state transition to hibernation whenever the system idle timer expires, enter:

   ```
   pmctrl −x −a hibernation
   ```

6. To set the system timer for suspend state at 17:00 on April 7, 1995, enter:

   ```
   pmctrl −S 9504071700 hibernation
   ```

7. To set the device idle time of hdisk0 to 2 minutes but not change the current device standby time, enter:

```
pmctrl −d hdisk0 −t 2 −1
```

Files

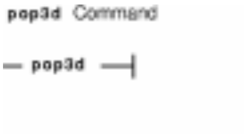
/usr/bin/pmctrl Contains the **pmctrl** command.

pop3d Daemon

Purpose

Starts the Post Office Protocol Version 3 (POP3) server process. This command only applies to AIX Version 4.2.1 or later.

Syntax



pop3d

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

None

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

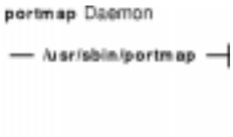
Daemons: **imapd**

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 Version 4.3 System Management Concepts: Operating System and Devices*.

System Resource Controller Overview in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

TCP/IP Protocols in *AIX Version 4.3 System Management Concepts: Operating System and Devices*.

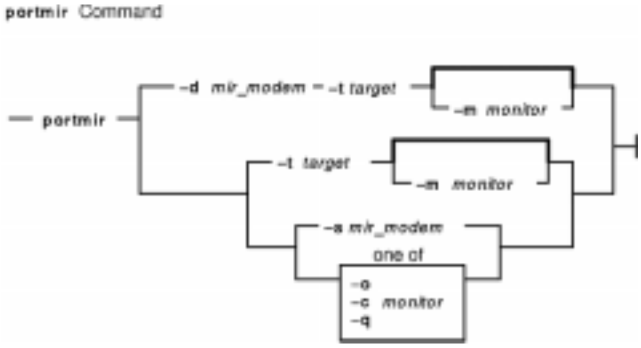
List of NFS Commands.

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. This command only applies to AIX Version 4.2.1 and later.

Syntax



portmir { **−d***mir_modem***−t** *target* [**−m***monitor*]|**−t** *target* [**−m** *monitor*] |**−s***mir_modem* } | { **−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. It is an enhancement of the **mirrord** command available on Micro Channel Adapter SMP machines with AIX Version 4.2.0 or earlier. The **portmir** command is available on all types of machines with AIX 4.2.1 or later and is backward−compatible with previous versions of the **mirrord** command.

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 Web−based System Manager Devices application (**wsm devices** fast path) to run this command. You could also use the System Management Interface Tool (SMIT) **smit portmir** fast path to run this command.

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.

−s *mir_modem* To be used in service mode only for Micro Channel Adapter SMP machines. For further information, refer to the **mirrord** daemon.

−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 **mirrord** daemon, the **chdev** command.

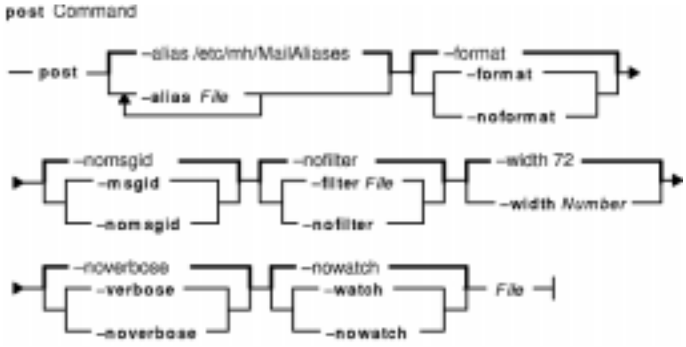
Setting up and running Web−based System Manager in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

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 Version 4.3 System User's Guide: Communications and Networks*.

Peek, Jerry. *MH and xmh: E−mail for Users and Programmers.* Sebastopol, CA: O'Reilly & Associates, 1992.

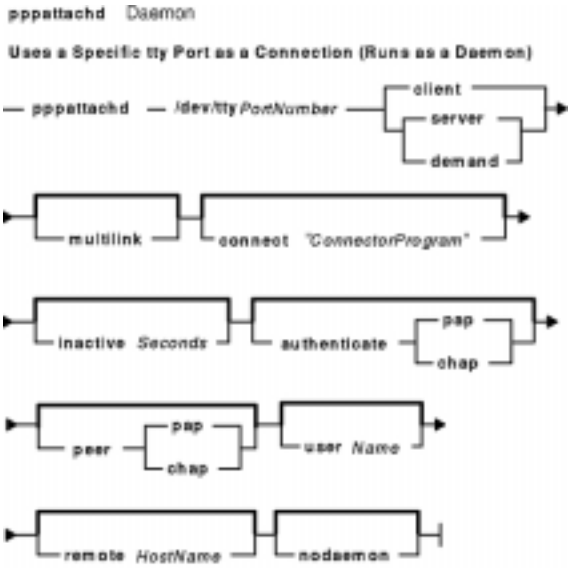
pppattachd Daemon

Purpose

Attaches an asynchronous device stream to the PPP (Point to Point Protocol) subsystem. Can be envoked as a daemon or a normal process.

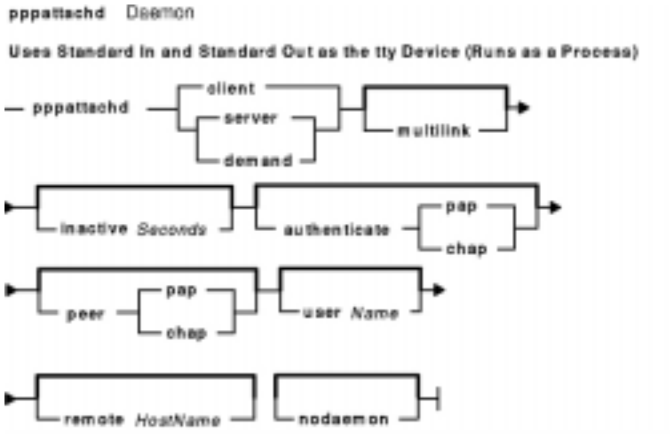
Syntax

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 exec'd 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.

In AIX Version 4.2.1 or later, 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. This option only applies to AIX Version 4.2.1 or later.

nodaemon
Specifies to the attachment process that it is not to become a daemon. You must use this option for attachment processes which 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 *UserNameRemoteHostNamePassword* 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 *UserNameRemoteHostNamePassword* 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 which 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 which 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 which 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 which the attachment was bound to. 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 Version 4.3 System Management Guide: Communications and Networks*.

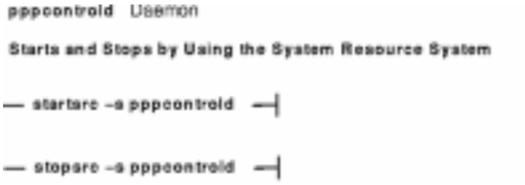
pppcontrold Daemon

Purpose

Controls startup and management of the PPP (Point to Point Protocol) subsystem.

Syntax

To Start and Stop by Using the System Resource Controller:



pppcontrold Usemon

Starts and Stops by Using the System Resource System

— startsrc –s pppcontrold —|

— stopsrc –s pppcontrold —|

startsrc −spppcontrold

stopsrc −spppcontrold

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 Version 4.3 System Management Concepts: Operating System and Devices*.

The System Resource Controller Overview in *AIX Version 4.3 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. This flag applies only to AIX Version 4.2 or later.
−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. This flag applies only to AIX Version 4.2 or later.
−v	Logs the chat activity using the syslog facility. This flag applies only to AIX Version 4.2 or later.

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 AIX 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 Version 4.3 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 Version 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**

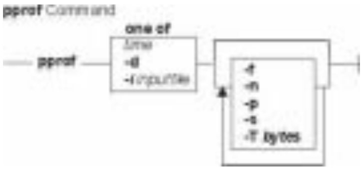
Asynchronous Point−to−Point Protocol (PPP) Subsystem Overview in *AIX Version 4.3 System Management Guide: Communications and Networks*.

pprof Command

Purpose

Reports CPU usage of all kernel threads over a period of time.

Syntax



pprof { *time* | **−i***input_file* | **−d** } [**−T***bytes*] [**−s**] [**−n**] [**−f**] [**−p**]

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 5 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.

−i*input_file* Indicates to generate reports from a previously generated **Pprof.flow**.

−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.

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.

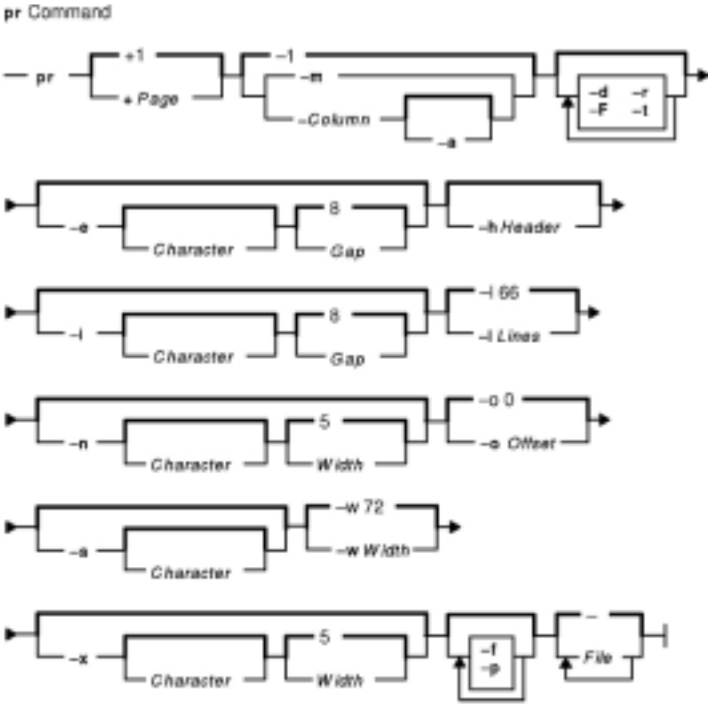
The **trcon** subroutine, and **trcstop** command.

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 multicolumns are filled horizontally, 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, enter:

   ```
   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, enter:

   ```
   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, enter:

```
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, enter:

   ```
   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 Version 4.3 System User's Guide: Operating System and Devices* describes files, file types, and how to name files.

Input and Output Redirection Overview in *AIX Version 4.3 System User's Guide: Operating System and Devices* describes how the operating system processes input and output.

National Language Support Overview for Programming in *AIX Version 4.3 General Programming Concepts: Writing and Debugging Programs* explains collating sequences, equivalence classes, and locale.

Shells Overview in *AIX Version 4.3 System User's Guide: Operating System and Devices* describes shells, the different types, and how they affect the way commands are interpreted.

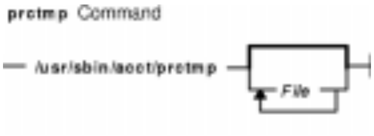
File and Directory Access Modes in *AIX Version 4.3 System Management Guide: Operating System and Devices* introduces file ownership and permissions to access files and directories.

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 Version 4.3 System Management Guide: Operating System and Devices*.

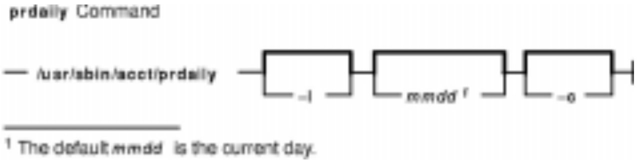
Setting Up an Accounting System in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

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 Version 4.3 System Management*

Guide: Operating System and Devices.

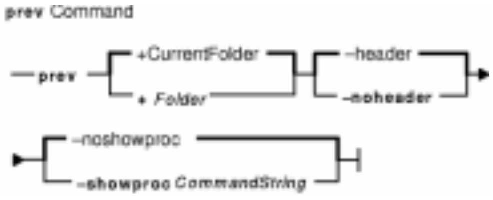
Setting Up an Accounting System in *AIX Version 4.3 System Management Guide: Operating System and Devices* explains the steps you must take to establish an accounting system.

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 Version 4.3 System User's Guide: Communications and Networks*.

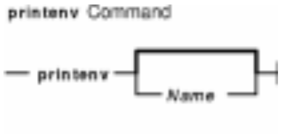
Peek, Jerry. *MH and xmh: E−mail for Users and Programmers.* Sebastopol, CA: O'Reilly & Associates, 1992.

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 Version 4.3 System Management Guide: Operating System and Devices*.

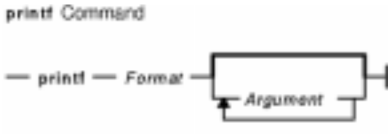
Shells Overview in *AIX Version 4.3 System User's Guide: Operating System and Devices*.

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 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 appear, 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 appear, 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.

   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 appear for the **d**, **o**, **i**, **u**, **x**, or **X** conversions.
   - The number of digits to appear 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 appear. 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 appear. 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 appear. 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 appear. 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 appears.

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 appears. 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 appears 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 *Technical Reference: Base Operating System and Extensions*.

The Input and Output Handling Programmer's Overview in *AIX General Programming Concepts: Writing and Debugging Programs* describes the files, commands, and subroutines used for low−level, stream, terminal, and asynchronous I/O interfaces.

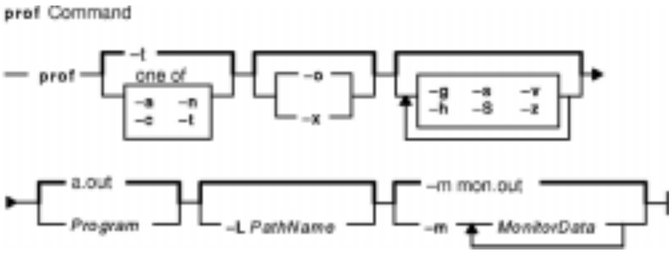
The National Language Support Overview for System Management in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

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 Version 4.3 System User's Guide: Operating System and Devices*.

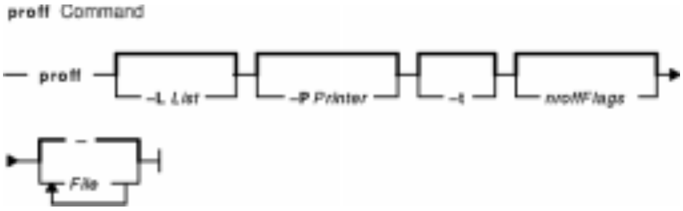
The Subroutines Overview in *AIX General Programming Concepts: Writing and Debugging Programs*.

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

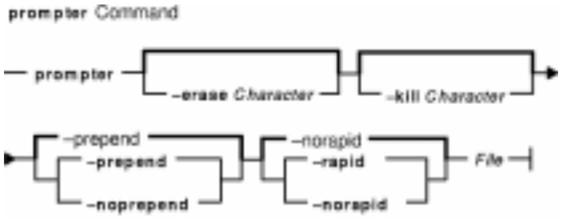
The **col** command, **eqn** command, **lpr** command, **nroff** command, **qprt** command, **tbl** command.

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 Version 4.3 System User's Guide: Communications and Networks*.

Peek, Jerry. *MH and xmh: E−mail for Users and Programmers.* Sebastopol, CA: O'Reilly & Associates, 1992.

proto Command

Purpose

Constructs a prototype file for a file system.

Syntax

proto Command
proto
Directory
Prefix

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 Overview for System Management in *AIX Version 4.3 System Management Guide: Operating System and Devices* explains file system types, management, structure, and maintenance.

The Files Overview in *AIX Version 4.3 System User's Guide: Operating System and Devices* provides information on working with files.

The National Language Support Overview for System Management in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

Locale Overview for System Management in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

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.

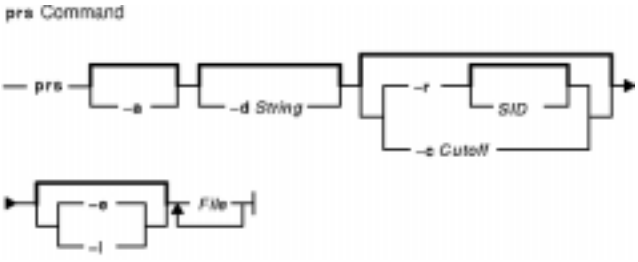
For an unmanaged proxy service, the proxy manager will look in the config file to find all unmanaged proxies for this service. If there is more than one entry in the config file with the same unmanaged *service−name*, the proxy manager will try to use the proxies in the order presented in the config 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 appear 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 appear 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

: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
:DI:	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
: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), enter:

   ```
   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`, enter:

   ```
   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.

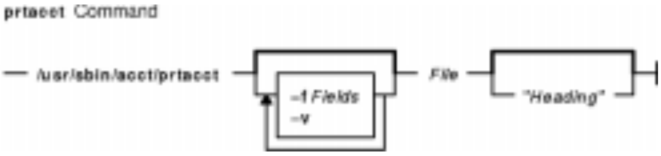
Source Code Control System (SCCS) Overview in *AIX General Programming Concepts: Writing and Debugging Programs*.

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 Version 4.3 System Management Guide: Operating System and Devices*.

Setting Up an Accounting System in *AIX Version 4.3 System Management Guide: Operating System and Devices* explains the steps you must take to establish an accounting system.

prtty Command

Purpose

Prints to the printer port of the terminal.

Syntax

prtty [**-l***Number*] [*File* ...]

Description

The **prtty** command prints the files specified by the *File* parameter to the printer port of your terminal. The **prtty** command reads the **TERM** environment variable and looks for that terminal in the **/usr/lib/INed/terms.bin** terminal description file. It uses the terminal description file to get the **k2** (enable printer port) and **k3** (disable printer port) sequences. If you do not specify a file, the editor reads from standard input.

Flags

-l*Number* Prompts you to start printing, prints number of lines specified with the *Number* variable, then prompts you again for further commands.

Related Information

INed Files.

The TERM Values for Different Displays and Terminals in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

Editor Overview in *AIX Version 4.3 INed Editor User's Guide* introduces general concepts about the INed editor.

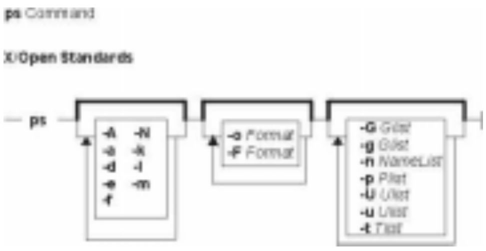
ps Command

Purpose

Shows current status of processes.

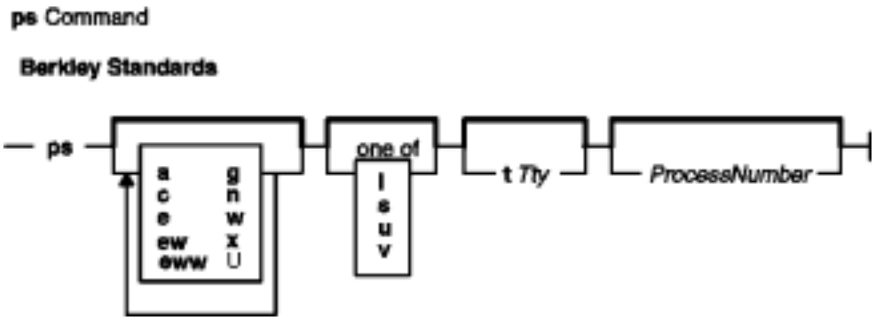
Syntax

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*]

Berkley 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 appear above the information displayed to standard output. The headings are defined in the following list (flags that cause these headings to appear 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.
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 below:

F Field Table		
Flags	**Hex 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 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 since several threads in the same porcess 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 seperated list of class names or a list of class names enclosed in double quotation marks (" "), which is seperated 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** simply 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 AIX operating system does not use the **−n** flag, since 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 **−ouser=** 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`.

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.

thcount

Indicates the number of kernel threads owned by the process. The default header for this field is THCNT.

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`.

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	**Field**	**Default**
Descriptors	**Specifiers**	**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 with no terminal.

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 specific users, enter:

   ```
   ps −f −l −ujim,jane,su
   ```

3. To list processes that are associated with the **/dev/console** and **/dev/tty1** ttys, enter:

   ```
   ps −t console,tty/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=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, enter:

   ```
   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 , enter:

   ```
   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, enter:

   ```
   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.

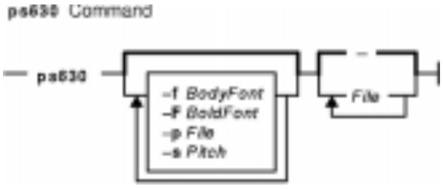
Using ps to Identify CPU−Intensive Programs in *AIX Versions 3.2 and 4 Performance Tuning Guide*.

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.

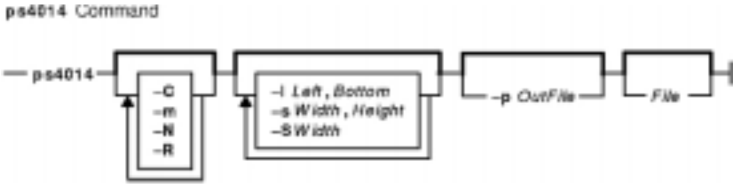
The **NLSvec** file.

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

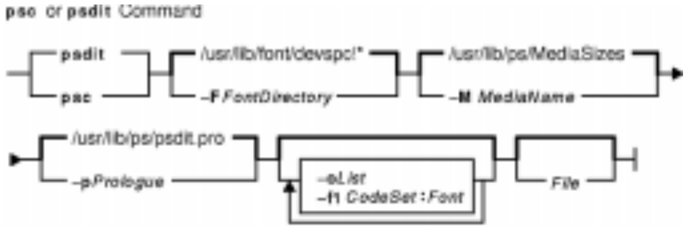
NLSvec file.

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 quotes. 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 quote 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	

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.

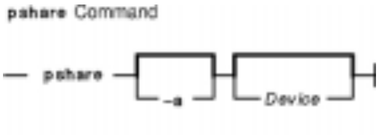
me Macro Package for the nroff and troff Commands.

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

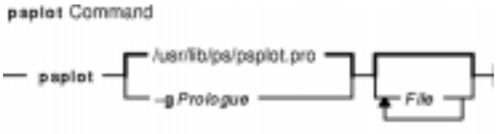
The **getty** command, **init** command, **pdelay** command, **pdisable** command, **penable** command, **phold** command, **pstart** command.

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.

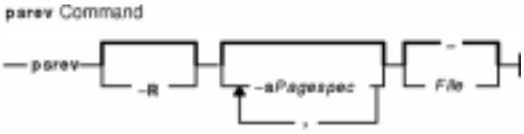
NLSvec File.

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

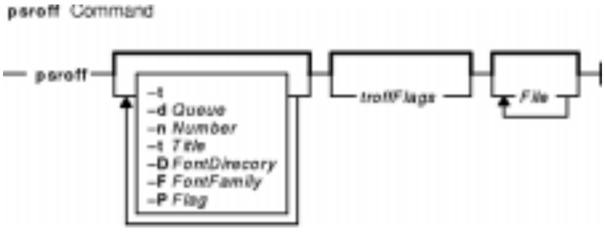
The **enscript** command.

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
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

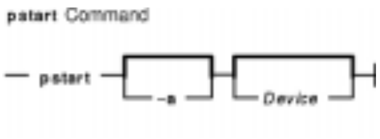
The **col** command, **enscript** command, **eqn** command, **lp** command, **managefonts** command, **nroff** command, **pic** command, **psdit** command, **refer** command, **tbl** command, **troff** command.

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.

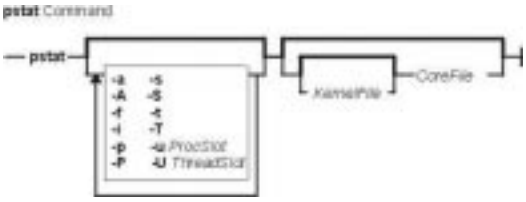
The **inittab** file.

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** command is a non−interactive form of the **crash** command. **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

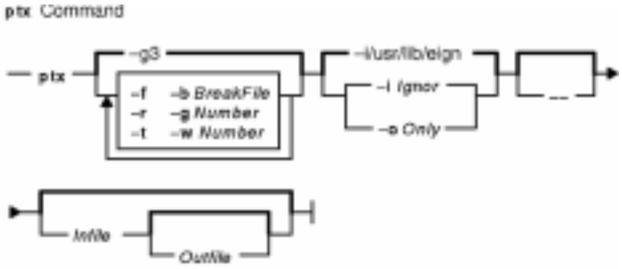
The **crash** command, **ps** command, **stty** command.

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.

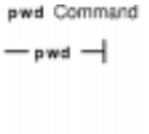
The **mm** macro package, **mptx** macro package**.**

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.

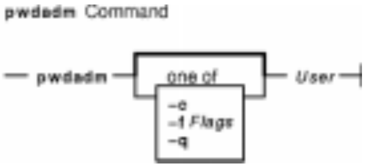
Directory Overview in *AIX Version 4.3 System User's Guide: Operating System and Devices*.

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 **−fNOCHECK** 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 **ADMCHGflags** 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.

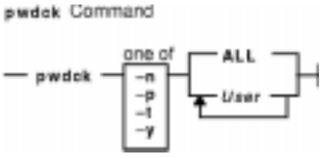
For more information about the identification and authentication of users, discretionary access control, the trusted computing base, and auditing, refer to Security Administration in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

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
PASSWORD_Checkerr	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.

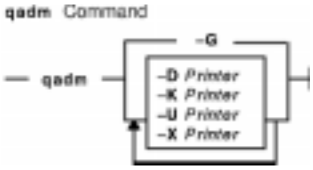
Security Administration in *AIX Version 4.3 System Management Guide: Operating System and Devices* describes the identification and authentication of users, discretionary access control, the trusted computing base, and auditing.

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 Web−based System Manager Printer Queues application (**wsm printers** fast path) to run this command. 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 Version 4.3 Guide to Printers and Printing*.

Spooler Overview for System Management in the *AIX Version 4.3 Guide to Printers and Printing*.

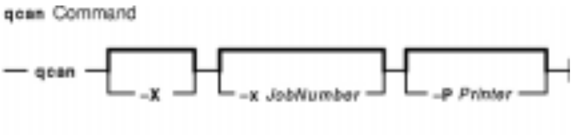
Setting up and running Web−based System Manager in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

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 Web−based System Manager Printer Queues application (**wsm printers** fast path) to run this command. 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 Version 4.3 System User's Guide: Operating System and Devices*.

Spooler Overview for System Management in the *AIX Version 4.3 Guide to Printers and Printing*.

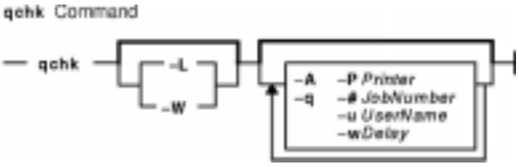
Setting up and running Web−based System Manager in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

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 Web−based System Manager Printer Queues application (**wsm printers** fast path) to run this command. 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 Version 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 Version 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 Version 4.3 Guide to Printers and Printing*.

The **enq** command, **qadm** command, **qcan** command, **qpri** command, **qprt** command.

The **/etc/qconfig** file.

Setting up and running Web−based System Manager in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

Checking Print Job Status (qchk Command) in the *AIX Version 4.3 Guide to Printers and Printing*.

qdaemon Command

Purpose

Schedules jobs enqueued by the **enq** command.

Syntax

qdaemon Command
qdaemon
1
1 This command is not usually entered on the command line.

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 Version 4.3 Guide to Printers and Printing*.

Queuing System Overview for System Management in the *AIX Version 4.3 Guide to Printers and Printing*.

National Language Support Overview for Programming in *AIX Version 4.3 General Programming Concepts: Writing and Debugging Programs*.

System Resource Controller Overview in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

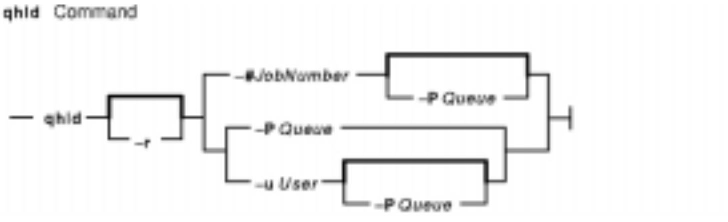
Understanding the Interaction between qdaemon and the Backend in the *AIX Version 4.3 Guide to Printers and Printing*.

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 Version 4.3 Guide to Printers and Printing*.

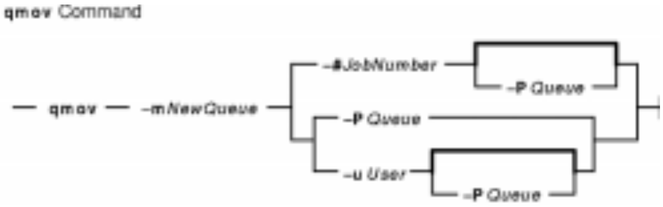
Spooler Overview for System Management in the *AIX Version 4.3 Guide to Printers and Printing*.

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 Version 4.3 Guide to Printers and Printing*.

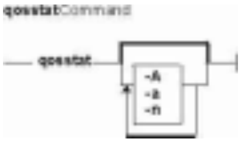
Spooler Overview for System Management in the *AIX Version 4.3 Guide to Printers and Printing*.

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

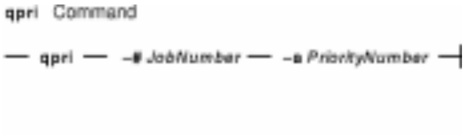
TCP/IP Quality of Service (QoS) in the *AIX Version 4.3 System Management Guide: Communications and Networks*.

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 Web−based System Manager Printer Queues application (**wsm printers** fast path) to run this command. 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 Version 4.3 System User's Guide: Operating System and Devices*.

Setting up and running Web−based System Manager in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

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 Web−based System Manager Printer Queues application (**wsm printers** fast path) to run this command. 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 Version 4.3 System Management Guide: Operating System and Devices*. Alternatively, consult your printer manual to find out what options your printer supports.

Flags

−#{j|h|v}
: 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. This flag applies to AIX Version 4.2 or later.

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 seperated 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−seperated 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 850, IBM−850, and 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 fime 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, Virtual Printer Definitions and Attributes, Initial Printer Configuration, Starting a Print Job (qprt Command) in *AIX Version 4.3 Guide to Printers and Printing*.

How to Print from a Remote System and How to Print to a Remote System in *AIX Version 4.3 System User's Guide: Communications and Networks*.

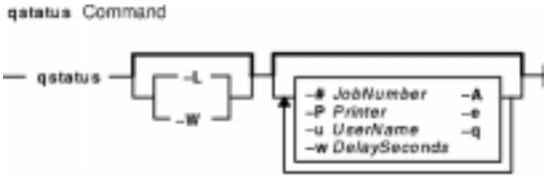
Setting up and running Web−based System Manager in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

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 Version 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 Version 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 Version 4.3 Guide to Printers and Printing*.

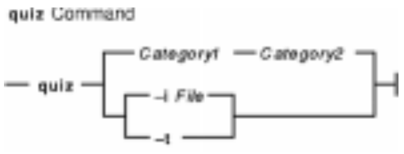
Remote Printing Overview in the *AIX Version 4.3 Guide to Printers and Printing*.

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 **a|ab** 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

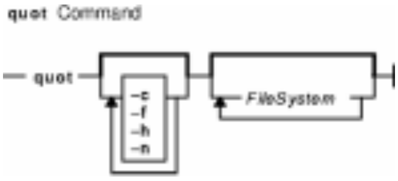
The **arithmetic** command, **back** command, **bj** command, **craps** command, **fish** command, **fortune** command, **hangman** command, **moo** command, **number** command, **ttt** command, **turnoff** command, **turnon** command, **wump** command.

quot Command

Purpose

Summarizes file system ownership.

Syntax



quot [**−c**] [**−f**] [**−h**] [**−n**] [*FileSystem ...*]

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

−c
Displays a 3−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 +0n | quot −n filesystem
```

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
```

Files

/etc/passwd Contains user names.
/etc/filesystems Contains file system names and locations.

Related Information

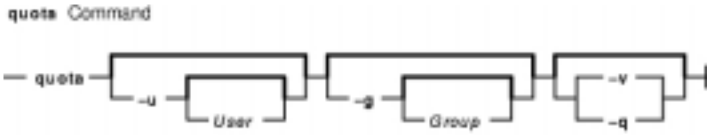
The **ls** command, **du** command.

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 may use the **−u** flag with the optional *User* parameter to view the limits of other users. Users without root user authority can view the limits of groups of which they are members by using the **−g** flag with the optional *Group* parameter.

> **Note:** 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`, enter:

```
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`, enter:

   ```
   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

How to Set Up the Disk Quota System in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

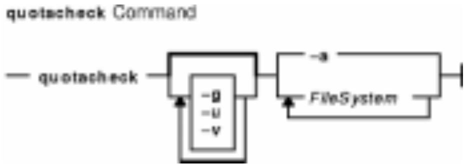
The **repquota** command summarizes quotas for a file system. The **quotacheck** command checks file system quota consistency. The **quotaon** and **quotaoff** commands enable and disable disk quotas, respectively. The **edquota** command edits user and group quotas.

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 normally 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, enter:

   ```
   quotacheck /usr
   ```

2. To check only the group quotas in the `/usr` file system, enter:

   ```
   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 **quota** command displays disk usage and quotas. The **repquota** command summarizes quotas for a file system. The **quotaon** and **quotaoff** commands enable and disable disk quotas, respectively. The **edquota** command edits user and group quotas.

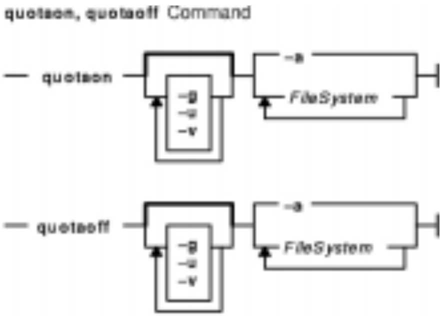
How to Set Up the Disk Quota System in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

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 displays disk usage and quotas. The **edquota** command creates and edits quotas. The **repquota** command summarizes quotas for a file system. The **quotacheck** command checks file system quota consistency.

The Disk Quota System Overview in *AIX Version 4.3 System Management Guide: Operating System and Devices* introduces the disk quota system.

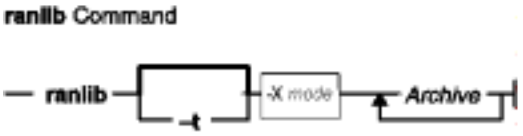
Setting Up the Disk Quota System in *AIX Version 4.3 System Management Guide: Operating System and Devices* describes how to establish disk quotas.

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 General Programming Concepts: Writing and Debugging Programs*.

The **ld** command, **ar** command, **lorder** command, **make** command.

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 system daemons such as the **cron** daemon are started by the **rc** command.
2. The root file system is implicitly mounted.

Related Information

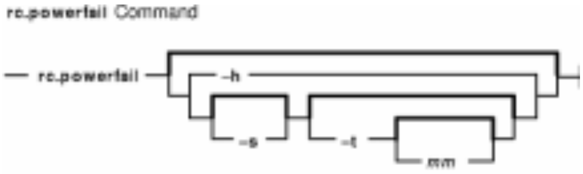
The **cron** daemon, **fsck** command, **init** command, **mount** command, **penable** command.

rc.powerfail Command

Purpose

Shuts down a system when a power failure is detected.

Syntax



rc.powerfail [**−h**] | [[**−s**] [**−t** [*mm*]]]

Description

The **rc.powerfail** command is started by the **/etc/inittab** file when **init** receives a SIGPWR signal from the kernel. The **rc.powerfail** command 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 critical power failure without any form of backup power supply, the **rc.powerfail** command is given approximately 20 seconds prior to shut down of the system. These critical situations are 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.

If there is a fan fault or if a battery backup power supply is attached to the system, the **rc.powerfail** command is given approximately 15 minutes prior to 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 at any time in the last 60 seconds the fan fault clears, the system shutdown halts and the users are notified that all errors have cleared. If a shutdown is NOT desired, the user may add the **−s** option to the command in the **/etc/inittab** file.

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

   ```
   rc.powerfail −h
   ```

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

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

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

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

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

The next SIGPWR received by **init** will have a 30 minute delay until system shutdown if the condition is not critical.

Files

/etc/inittab Controls the initialization process.

Related Information

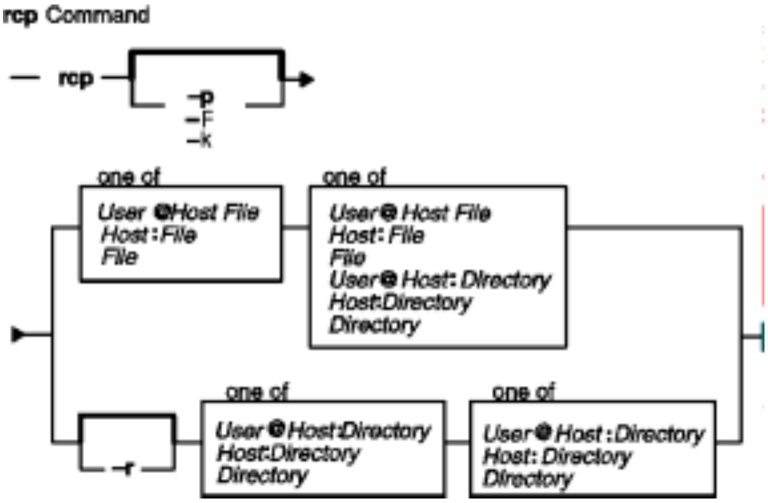
The **machstat** command, **init** command, **chitab** command, **halt** command, **shutdown** command.

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. Normally, 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 AIX 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 non−AIX 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.

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.

−FCauses 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.

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.

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.

Related Information

The **cp** command, **ftp** command, **rlogin** command, **rsh** command, **telnet** command, **tftp** command, **umask** command.

The **rshd** and **krshd** daemon.

The **kvalid_user** function.

The **.rhosts** file format, **hosts.equiv** file format.

Network Overview in *AIX Version 4.3 System Management Guide: Communications and Networks*.

Secure Rcmds in *AIX Version 4.3 System User's Guide: Communications and Networks*.

rcvdist Command

Purpose

Sends a copy of incoming messages to additional recipients.

Syntax

rcvdist Command
rcvdist
−form rcvdistcomps
−form File
User

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 Version 4.3 System User's Guide: Communications and Networks*.

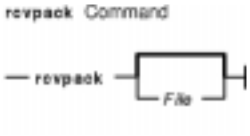
Peek, Jerry. *MH and xmh: E−mail for Users and Programmers.* Sebastopol, CA: O'Reilly & Associates, 1992.

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 Version 4.3 System User's Guide: Communications and Networks*.

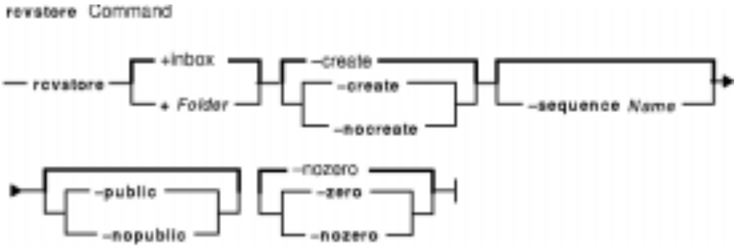
Peek, Jerry. *MH and xmh: E−mail for Users and Programmers.* Sebastopol, CA: O'Reilly & Associates, 1992.

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 Version 4.3 System User's Guide: Communications and Networks*.

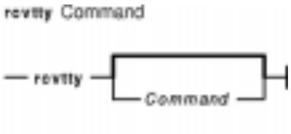
Peek, Jerry. *MH and xmh: E−mail for Users and Programmers.* Sebastopol, CA: O'Reilly & Associates, 1992.

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 Version 4.3 System User's Guide: Communications and Networks*.

Peek, Jerry. *MH and xmh: E−mail for Users and Programmers.* Sebastopol, CA: O'Reilly &

Associates, 1992.

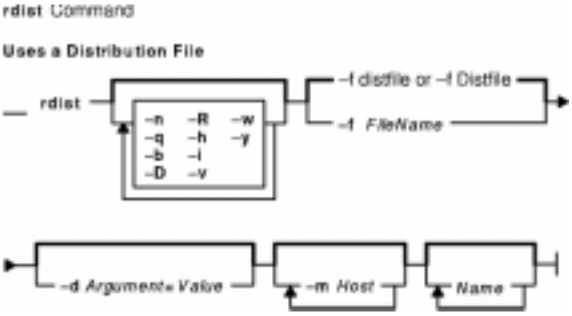
rdist Command

Purpose

Distributes identical copies of files on multiple hosts.

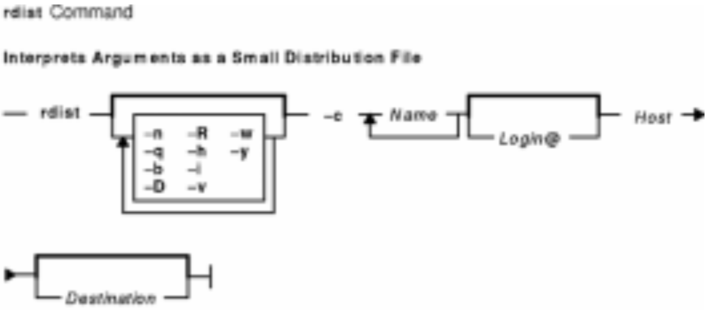
Syntax

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 **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 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.

−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 Specifies that the command completed successfully.

>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 Version 4.3 System Management Guide: Communications and Networks*.

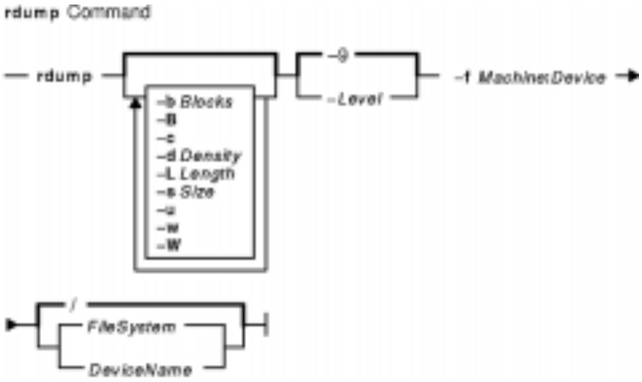
rdump Command

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, enter `−L 2g`. This flag only applies to AIX Version 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 normally 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, enter:

   ```
   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, enter:

   ```
   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, enter:

   ```
   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.

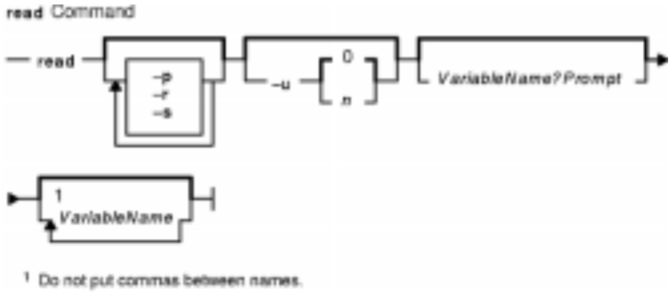
Backup Overview for System Management, Directory Overview, Files Overview, File Systems Overview for System Management in *AIX Version 4.3 System User's Guide: Operating System and Devices*.

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, enter:

   ```
   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, enter:

   ```
   (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, enter:

   ```
   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.

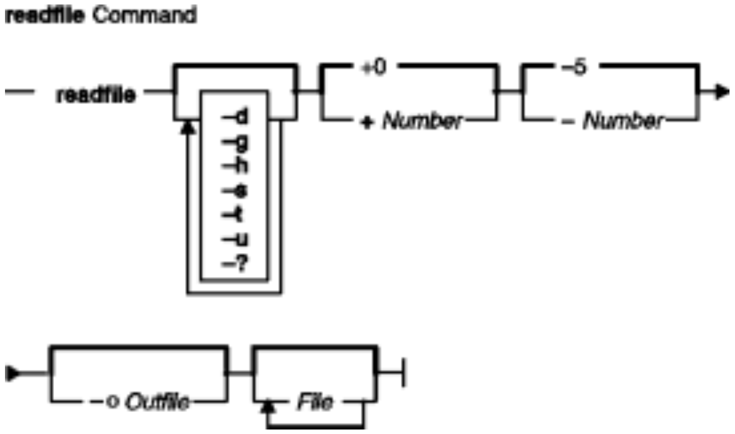
The **ksh** command.

readfile Command

Purpose

Displays the text of INed structured files.

Syntax



readfile [**−d**] [**−g**] [**−h**] [**−s**] [**−t**] [**−u**] [**−?**] [**+***Number*] [**−***Number*]
[**−o** *Outfile*] [*File* ...]

Description

The **readfile** command reads the specified structured file or a list of structured files and creates an ASCII printout to illustrate the file's tree structure. If the file is a structured file, the **readfile** command only prints the file's string data; there is no indentation. If the file is a text file, the **readfile** command works like the **cat** command.

If you do not specify the *File* parameter, the **readfile** command reads standard input as a text file.

Flags

−d Creates a detailed formatted dump of the contents of the input files.

−g Converts the control characters, such as graphic characters used in forms, to similar printing characters.

−h Separates the output into sections with headers that identify the input file for each section.

−s Suppresses messages that report unstructured and nonexistent files.

−t Illustrates the structure of the file with a tree diagram. (The editor ignores this flag if the file is a structured file or a text file.)

−u Suppresses buffering of output. (The default buffer size is the size of the disk block.)

−? Displays a usage syntax message.

+*Number* Begins reading the file at the record specified by the *Number* variable. (The first record is record 0.) The default for this flag is +0. Separate this flag from the other flags with a space.

−*Number*	For structured files, sets the size of the increments and decrements for indentation at the specified number of columns. This flag is ignored if you specify the **−t** flag. Separate this flag from the other flags with a space.
−o *Outfile*	Uses the value of the *Outfile* variable as the output file instead of standard output. If the output file name is the same name as an input file, the editor backs up the input file in a **.bak** file.

Examples

1. To view your editor profile with headings, enter:

   ```
   readfile −h profiles/editorprf
   ```

2. To change the `stextfile` structured file to the `asciifile` text file, enter:

   ```
   readfile stextfile>asciifile
   ```

Related Information

The **cat** command.

INed Editor Overview in *AIX Version 4.3 INed Editor User's Guide* introduces general concepts about the INed editor.

How to Access Previous Versions of a File with the INed Editor.

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

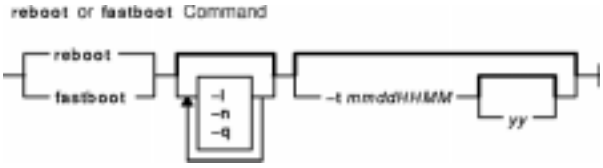
The **chlvcopy** command.

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 **false**. To reset the **autorestart** attribute value to **true**, use the **chdev** command and enter:

```
chdev −l sys0 −a autorestart=true.
```

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.

−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.

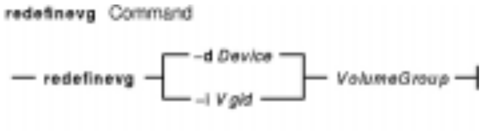
The **utmp**, **wtmp**, **failedlogin** File Format.

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 Command
reducevg
−d
−f
VolumeGroup
PhysicalVolume

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***VolumeGroupPhysicalVolume*. 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***VolumeGroupPVID*

You can use the Web−based System Manager Volumes application (**wsm lvm** fast path) to run this command. 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 Version 4.3 System Management Guide: Operating System and Devices*.

Setting up and running Web−based System Manager in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

The System Management Interface Tool (SMIT) Overview in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

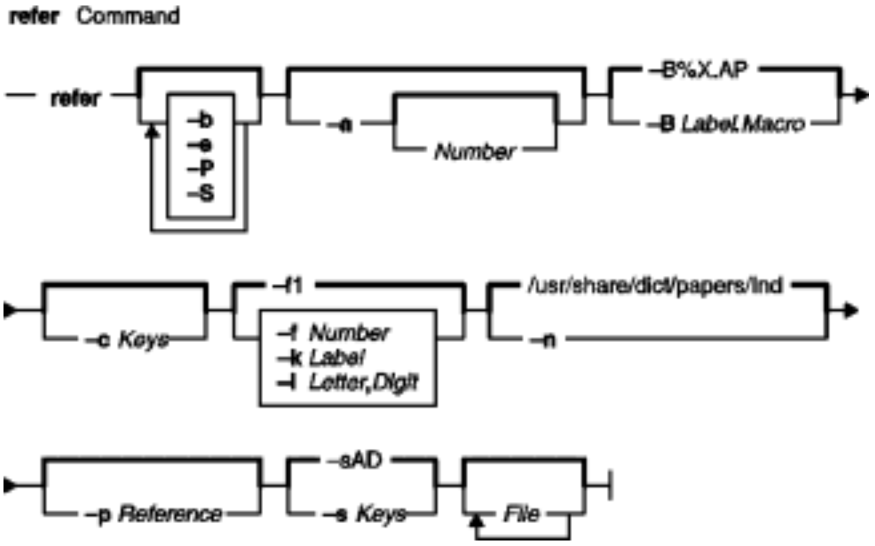
AIX HACMP/6000 Concepts and Facilities.

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/Ind Contains the default reference file.

/usr/lbin/refer Contains companion programs.

Related Information

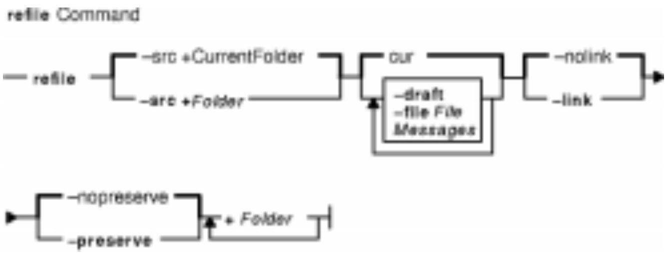
The **eqn** command, **addbib** command, **indxbib** command, **lookbib** command, **neqn** command, **nroff** command, **roffbib** command, **sortbib** command, **tbl** command, **troff** command.

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.

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 Version 4.3 System User's Guide: Communications and Networks*.

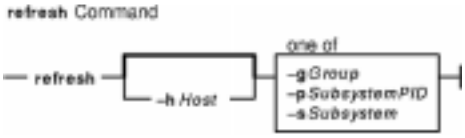
Peek, Jerry. *MH and xmh: E−mail for Users and Programmers.* Sebastopol, CA: O'Reilly & Associates, 1992.

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

To refresh the `tcpip` subsystem, enter:

```
refresh −s tcpip
```

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

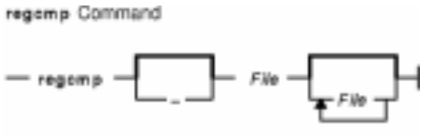
The System Resource Controller Overview in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

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.

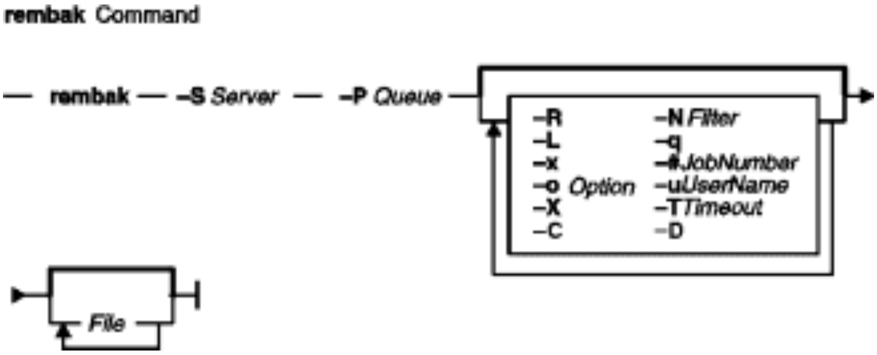
The Subroutines Overview in *AIX General Programming Concepts: Writing and Debugging Programs*.

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 a 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 AIX system. The **lpd** daemon in AIX 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 a 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 Version 4.3 Guide to Printers and Printing*.

Spooler Overview for System Management in the *AIX Version 4.3 Guide to Printers and Printing*.

Remote Printing Overview in the *AIX Version 4.3 Guide to Printers and Printing*.

Using Remote Host Access for Printing in the *AIX Version 4.3 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 Version 4.3 System Management Guide: Operating System and Devices*.

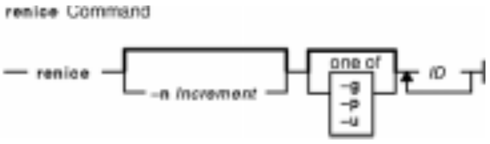
Setting Up an Accounting System in *AIX Version 4.3 System Management Guide: Operating System and Devices* explains the steps you must take to establish an accounting system.

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 Version 4.3 System User's Guide: Operating System and Devices* describes how commands are processed.

The Processes Overview in *AIX Version 4.3 System User's Guide: Operating System and Devices* discusses what processes are and input and output redirection.

The Shells Overview in *AIX Version 4.3 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 Versions 3.2 and 4 Performance Tuning Guide*.

reorgvg Command

Purpose

Reorganizes the physical partition allocation for a volume group.

Syntax

reorgvg Command
reorgvg
-i
VolumeGroup
LogicalVolume

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.

   In AIX Version 4.2 or later, If you enter the **reorgvg** command with the volume group name and no other arguments, the entire volume group is reorgnized.

You can use the Web−based System Manager Volumes application (**wsm lvm** fast path) to run this command. 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 Version 4.3 System Management Guide: Operating System and Devices*.

Setting up and running Web−based System Manager in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

The System Management Interface Tool (SMIT) Overview in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

Monitoring and Tuning Disk I/O in *AIX Versions 3.2 and 4 Performance Tuning Guide*.

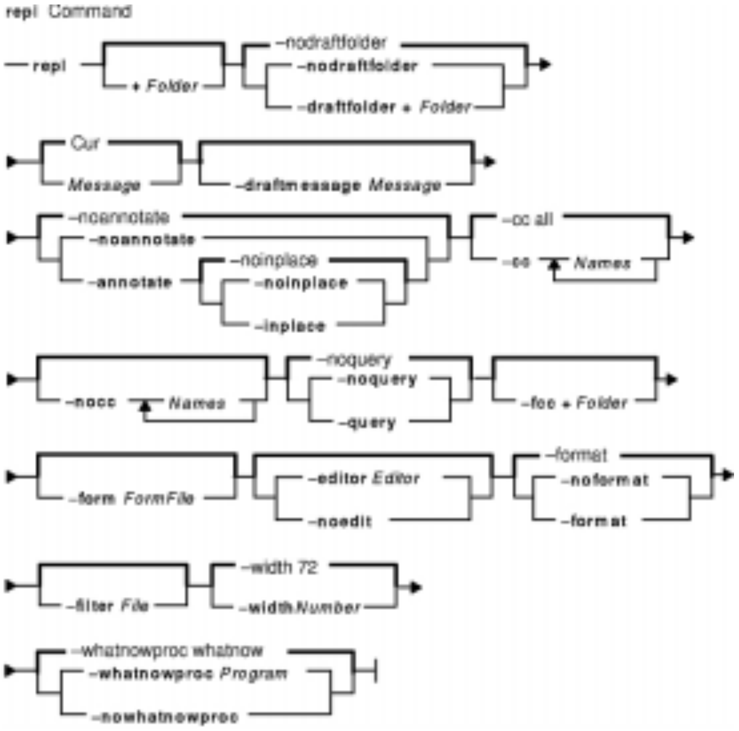
AIX HACMP/6000 Concepts and Facilities.

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 Version 4.3 System User's Guide: Communications and Networks*.

Peek, Jerry. *MH and xmh: E−mail for Users and Programmers.* Sebastopol, CA: O'Reilly & Associates, 1992.

replacepv Command

Purpose

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

Syntax

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

replacepv [**−R**] *dir_name* [*DestinationPhysicalVolume*]

Description

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

Note:

1. The *DestinationPhysicalVolume* 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.

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 Version 4.3 System Management Guide: Operating System and Devices*.

Setting up and running Web−based System Management in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

System Management Interface Tool (SMIT): Overview in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

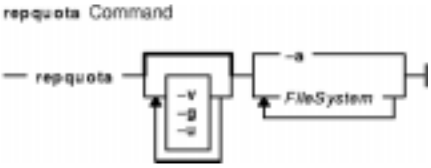
System Dump Facility in *AIX Version 4.3 Problem Solving Guide and Reference*.

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 Version 4.3 System Management Guide: Operating System and Devices*.

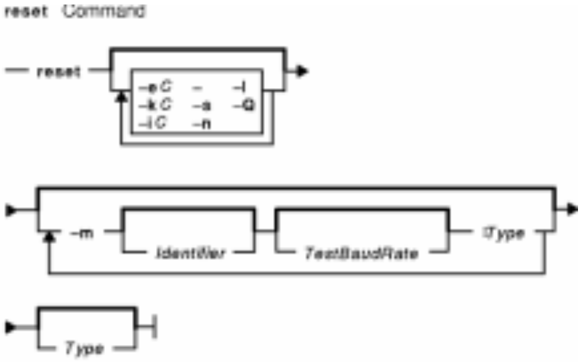
How to Set Up the Disk Quota System in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

reset Command

Purpose

Initializes terminals.

Syntax



reset [**−e***C*] [**−k***C*] [**−i***C*] [**−**] [**−s**] [**−n**] [**−l**] [**−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.

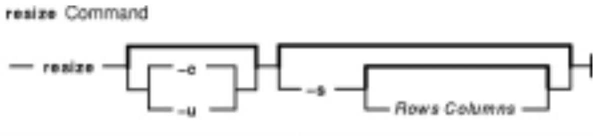
TERM Values for Different Displays and Terminals section of tty Overview for System Managers in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

resize Command

Purpose

Sets the **TERMCAP** environment variable and terminal settings to the current window size.

Syntax



resize [**−c** | **−u**] [**−s** [*RowsColumns*]]

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

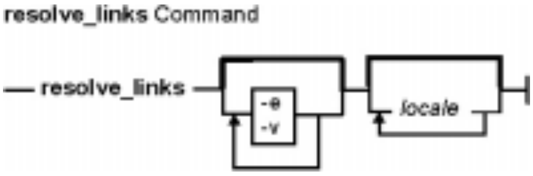
The **csh** command, **tset** command, **xterm** command.

resolve_links Command

Purpose

Enables and disables cross−book links to HTML library books that were not installed on the system.

Syntax



resolve_links [**−e**] [**−v**] [*locale* ...]

Description

The **resolve_links** command disables and enables HTML links in the AIX documentation to documents in other books.

The AIX documentation library allows administrators to install only some of the books in the library. Many of the books in the library have cross−book HTML links that take you to other books in the library. By default, if a cross−book link points to a book that is not installed or mounted from a documentation CD, the get a `File Not Found` error message if you click on the link. Use the **resolve_links** command to disable cross−book links to missing books. When a link is disabled, the text label for the link still appears, but it is not underlined, and does not act as a link.

By default, the **resolve_links** command modifies all documents that are installed in the **a_doc_lib** directory using the AIX installation procedures. This directory is found under the **/usr/share/man/info/***locale* directories (where *locale* is replaced by the names of all of the locales for which you have installed documentation). Optionally, you may include one or more locale names with the command so that only documents under the path(s) with the specified *locale*(s) are modified.

Flags

−e Enable all cross−book HTML links regardless of whether or not the target books are installed.
−v Verbose output.

Example

1. To disable cross−book HTML links to missing documents for all languages, enter:
   ```
   resolve_links
   ```
2. To disable cross−book HTML links in the Spanish library that point to missing books, enter:
   ```
   resolve_links es_ES
   ```
3. To enable all cross−book HTML links in the Spanish and Japanese libraries regardless of whether or not the target books are installed, enter:
   ```
   resolve_links −e es_ES Ja_JP
   ```

Files

/usr/sbin/resolve_links	Contains the **resolve_links** command.
/usr/share/man/info/*/a_doc_lib directories	Contain the AIX HTML documentation library files.

Related Information

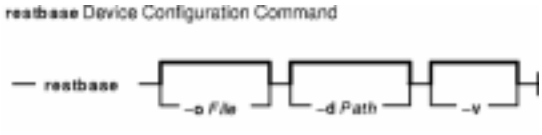
Locale Overview for System Management in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

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 Version 4.3 Kernel Extensions and Device Support Programming Concepts*.

Object Data Manager (ODM) Overview for Programmers in *AIX Version 4.3 General Programming Concepts: Writing and Debugging Programs*.

List of Device Configuration Commands in *AIX Version 4.3 Kernel Extensions and Device Support Programming Concepts*.

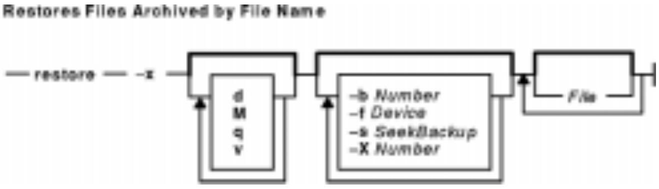
restore Command

Purpose

Extracts files from archives created with the **backup** command.

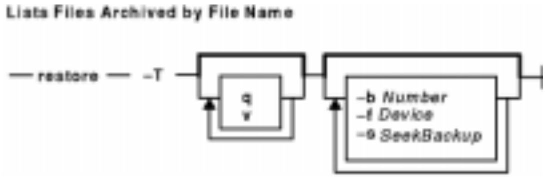
Syntax

To Restore Files Archived by File Name



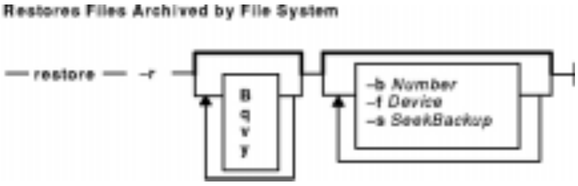
restore −x [**dM vq**] [**−b***Number*] [**−f***Device*] [**−s***SeekBackup*] [**−X***VolumeNumber*] [*File* ...]

To List Files Archived by File Name



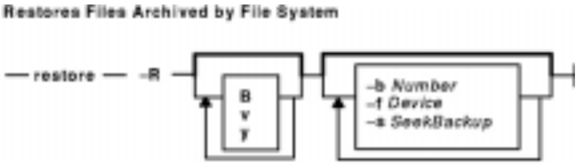
restore −T [**q v**] [**−b***Number*] [**−f***Device*] [**−s***SeekBackup*]

To Restore Files Archived by File System



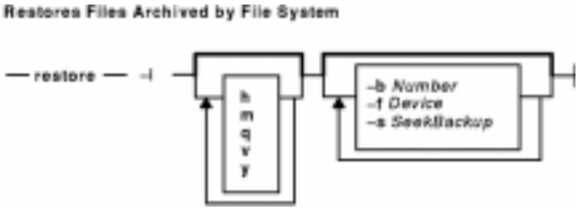
restore −r[**B qvy**] [**−b***Number*] [**−f***Device*] [**−s***SeekBackup*]

To Restore Files Archived by File System



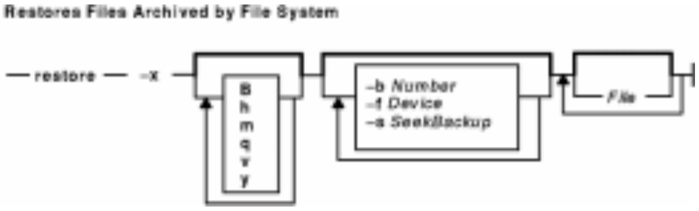
restore −R[**Bvy**] [**−b***Number*] [**−f***Device*] [**−s***SeekBackup*]

To Restore Files Archived by File System



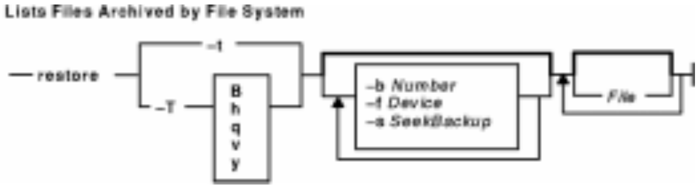
restore −i[**hmqvy**] [**−b** *Number*] [**−f** *Device*] [**−s** *SeekBackup*]

To Restore Files Archived by File System



restore −x[**Bhmqvy**] [**−b** *Number*] [**−f***Device*] [**−s***SeekBackup*] [*File* ...]

To List Files Archived by File System



restore −t| **−T**[**Bhqvy**] [**−b***Number*] [**−f***Device*] [**−s***SeekBackup*] [*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 (...).

Flags

−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 tape's physical block size. If the read size is not an even multiple of the tape's 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.

−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 Version 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.

−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 tape's read/write head and not to an archive's 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 output. 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 and a list of files found on the archive are 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 seperated 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**, enter:

   ```
   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, enter:

   ```
   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, enter:

   ```
   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, enter:

   ```
   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, enter:

   ```
   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, enter:

   ```
   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 archive's position on the tape. The ninth archive is the fourth archive from your current position on the tape.

7. To restore the fourth backup, which begins on the sixth tape on a 10−tape multiple−backup

archive, put the sixth tape into the tape drive and enter:

```
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 entering:

```
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 entering:

```
chdev −l DeviceName −a BlockSize=32k
```

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 Version 4.3 System Management Guide: Operating System and Devices*.

The File Systems Overview for System Management in *AIX Version 4.3 System Management Guide: Operating System and Devices* explains file system types, management, structure, and maintenance.

The System Management Interface Tool (SMIT) Overview in *AIX Version 4.3 System Management*

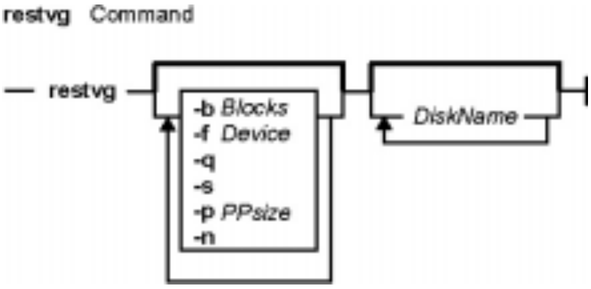
Guide: Operating System and Devices.

restvg Command

Purpose

Restores the user volume group and all its containers and files.

Syntax



restvg [**−b***Blocks*] [**−f***Device*] [**−q**] [**−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.

−f *Device* Specifies the device name of the backup media. The default is **/dev/rmt0**.

−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.

−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
```

Related Information

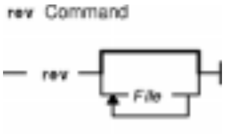
The mkvgdata command, **restore** command, **savevg** command, **mkdev** command, **reducevg** command.

rev Command

Purpose

Reverse 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, enter:

```
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

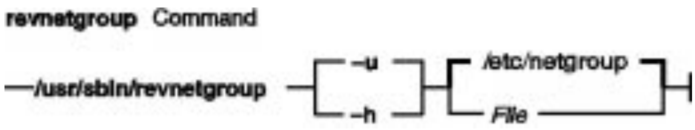
Input and Output Redirection Overview in *AIX Version 4.3 System User's Guide: Operating System and Devices*.

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 Version 4.3 System Management Guide: Communications and Networks*.

Network Information Service (NIS) in *AIX Version 4.3 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 Version 4.3 System Management Guide: Communications and Networks*.

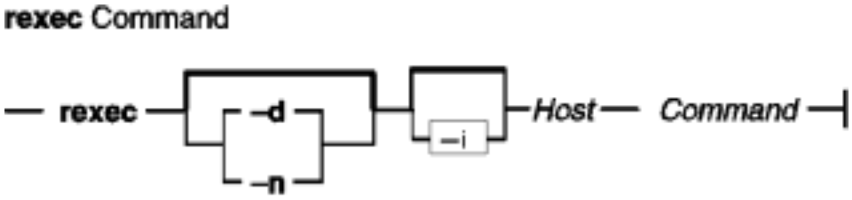
List of NFS Commands.

rexec Command

Purpose

Executes commands one at a time on a remote host.

Syntax



rexec [**−d** | **−n**] [**−i**] *HostCommand*

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

−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 −n host1 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 li −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 Version 4.3 System Management Guide: Communications and Networks*.

The **ftp** command, **rlogin** command, **rsh** command, **securetcpip** command.

The **rexecd** daemon.

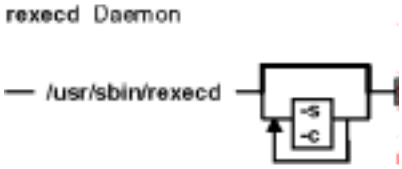
The **.netrc** file format.

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 Version 4.3 System Management Guide: Communications and Networks*.

Setting up and running Web−based System Manager in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

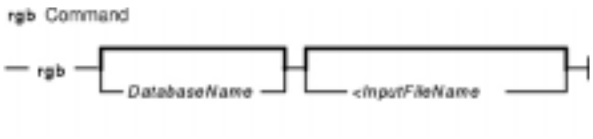
TCP/IP Daemons in *AIX Version 4.3 System Management Guide: Communications and Networks*.

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

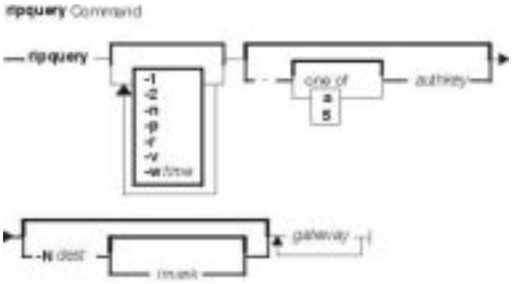
/usr/lib/X11/rgb.txt The default rgb database input file.

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

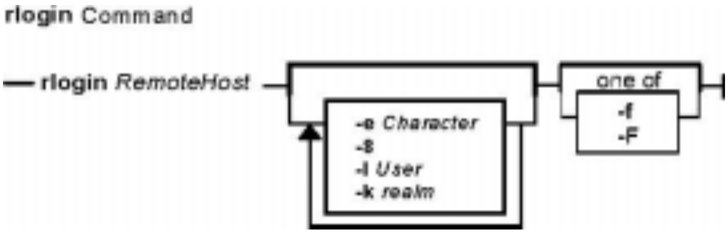
The **gated**Daemon.

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 AIX 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 non−AIX 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 Version 4.3 System Management Guide: Communications and Networks*.

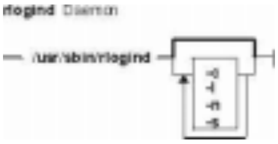
Secure Rcmds in *AIX Version 4.3 System User's Guide: Communications and Networks*.

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 [**−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

−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 Version 4.3 System Management Guide: Communications and Networks*.

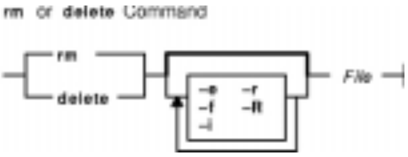
Setting up and running Web−based System Manager in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

rm or delete Command

Purpose

Removes (unlinks) files or directories.

Syntax



{ **rm** | **delete** } [**−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. 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 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 −ir manual
   ```

   This command recursively removes the contents of all subdirectories of the `manual` directory, prompting you regarding the removal of each file, and then removes the `manual` directory itself, for example:

   ```
   You:  rm −ir manual
   System: rm: Select files in directory manual? Enter y for yes.
   You:  y
   System: rm: Select files in directory manual/draft1? Enter y for yes.
   You:  y
   System: rm: Remove manual/draft1?
   You:  y
   System: rm: Remove manual/draft1/chapter1?
   You:  y
   System: rm: Remove manual/draft1/chapter2?
   You:  y
   System: rm: Select files in directory manual/draft2? Enter y for yes.
   You:  y
   System: rm: Remove manual/draft2?
   You:  y
   System: rm: Remove manual?
   ```

```
You:  y
```

Here, the **rm** command first asks if you want it to search the `manual` directory. Because the `manual` directory contains directories, the **rm** command next asks for permission to search `manual/draft1` for files to delete, and then asks if you want it to delete the `manual/draft1/chapter1` and `manual/draft1/chapter2` files. The **rm** command next asks for permission to search the `manual/draft2` directory. Then asks for permission to delete the `manual/draft1`, `manual/draft2`, and `manual` directories.

If you deny permission to remove a subdirectory (for example, `manual/draft2`), the **rm** command does not remove the `manual` directory. Instead, you see the message: `rm: Directory manual not empty`.

Files

/usr/bin/rm	Contains the **rm** command.
/usr/bin/delete	Contains the **delete** command.

Related Information

The **del** command, **ln** command.

The **unlink** subroutine.

Directory Overview in *AIX Version 4.3 System User's Guide: Operating System and Devices*.

File and Directory Access Modes in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

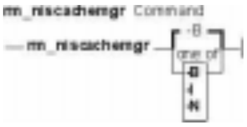
Linking Files and Directories in *AIX Version 4.3 System User's Guide: Operating System and Devices*.

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 Version 4.3 Network Information Services (NIS and NIS+) Guide*

Network File System (NFS) Overview for System Management in *AIX Version 4.3 Network Information Services (NIS and NIS+) Guide*

How to Start the NFS Daemons, How to Stop the NFS Daemons, How to Export a File System Using Secure NFS, How to Mount a File System Using Secure NFS in *AIX Version 4.3 Network Information Services (NIS and NIS+) Guide*

Setting up and running Web−based System Management in *AIX Version 4.3 Network Information Services (NIS and NIS+) Guide*

System Management Interface Tool (SMIT) Overview in *AIX Version 4.3 Network Information Services (NIS and NIS+) Guide*

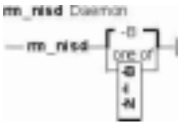
NIS+ Reference.

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 Version 4.3 Network Information Services (NIS and NIS+) Guide*

Network File System (NFS) Overview for System Management in *AIX Version 4.3 Network Information Services (NIS and NIS+) Guide*

How to Start the NFS Daemons, How to Stop the NFS Daemons, How to Export a File System Using Secure NFS, How to Mount a File System Using Secure NFS in *AIX Version 4.3 Network Information Services (NIS and NIS+) Guide*

Setting up and running Web−based System Management in *AIX Version 4.3 Network Information Services (NIS and NIS+) Guide*

System Management Interface Tool (SMIT) Overview in *AIX Version 4.3 Network Information Services (NIS and NIS+) Guide*

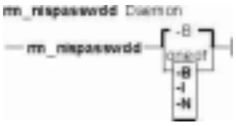
NIS+ Reference.

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 Version 4.3 Network Information Services (NIS and NIS+) Guide*

Network File System (NFS) Overview for System Management in *AIX Version 4.3 Network Information Services (NIS and NIS+) Guide*

How to Start the NFS Daemons, How to Stop the NFS Daemons, How to Export a File System Using Secure NFS, How to Mount a File System Using Secure NFS in *AIX Version 4.3 Network Information Services (NIS and NIS+) Guide*

Setting up and running Web−based System Management in *AIX Version 4.3 Network Information Services (NIS and NIS+) Guide*

System Management Interface Tool (SMIT) Overview in *AIX Version 4.3 Network Information Services (NIS and NIS+) 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 Version 4.3 System Management Guide: Communications and Networks*.

rmclass Command

Purpose

Remove a Workload Management class.

Syntax

rmclass [**−d** *Config_dir*] *Name*

Description

The **rmclass** removes the class identified by the *Name* parameter from the class definition file and the class limits file. The class must already exist. The predefined *Default* and *System* classes cannot be removed.

Flags

−d*Config_dir* Use */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.

Files

/etc/wlm/current/classes	Contains the names and definitions of the classes for the current workload management configuration.
/etc/wlm/current/limits	Contains the resource limits for the current workload management configuration.
/etc/wlm/current/shares	Contains the resource shares for the current workload management configuration.
/etc/wlm/current/description	Contains the class description text for each class of the current workload management configuration.

Related Information

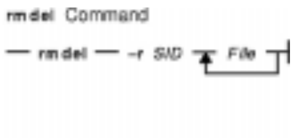
The **wlmcntrl**, **lsclass**, **chclass** and the **mkclass** commands.

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, enter:

```
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 General Programming Concepts: Writing and Debugging Programs*.

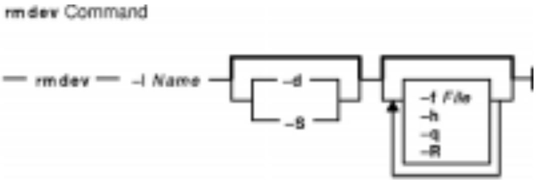
List of SCCS Commands in *AIX General Programming Concepts: Writing and Debugging Programs*.

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 Web−based System Manager Devices application (**wsm devices** fast path) to run this command. 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* variable.

−h Displays the command usage message.

−l *Name* Specifies the logical device, indicated by the *Name* variable, in the Customized Devices object class.

−q Suppresses the command output messages from standard output and standard error.

−R Unconfigures the children of the device. 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, enter:

   ```
   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, enter:

   ```
   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, enter:

   ```
   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 Version 4.3 System Management Guide: Operating System and Devices*.

Setting up and running Web−based System Manager in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

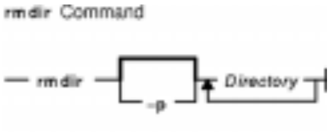
System Management Interface Tool (SMIT): Overview in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

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 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, enter:

   ```
   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 **rmmydir/* 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 normally list them unless you use the **−a** flag.

2. To remove the `/home`, `/home/demo`, and `/home/demo/mydir` directories, enter:

   ```
   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 Version 4.3 System User's Guide: Operating System and Devices*.

Directory Overview in *AIX Version 4.3 System User's Guide: Operating System and Devices*.

File and Directory Access Modes in *AIX Version 4.3 System User's Guide: Operating System and Devices*.

The **ls** command, **mkdir** command, **rm** command.

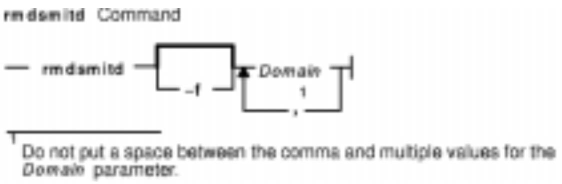
The **rmdir** subroutine, **unlink** subroutine.

rmdsmitd Command

Purpose

Removes domains from the member list in the Distributed System Management Interface Tool (DSMIT).

Syntax


rmdsmitd Command
rmdsmitd
−f
Domain
1
,
Do not put a space between the comma and multiple values for the Domain parameter.

rmdsmitd [**−f**] *Domain* [**,***Domain*] ...

> **Note:** Do not put a space between the comma and multiple values for the *Domain* parameter.

Description

The **rmdsmitd** command removes domains from the member list in DSMIT. The *Domain* parameter specifies the domains to be removed. The **rmdsmitd** command removes only empty domains. To remove nonempty domains, specify the **rmdsmitd** command with the **−f** flag.

Flags

−? Displays the usage statement.

−f Removes all the specified domains, including nonempty ones.

Examples

1. To remove the empty `domain_1` domain, enter:

   ```
   rmdsmitd domain_1
   ```

2. To remove a nonempty `domain_1` domain, enter:

   ```
   rmdsmitd −f domain_1
   ```

Files

/usr/share/DSMIT/domains Contains the list of domains used by DSMIT.

Related Information

The **chdsmitd** command, **mkdsmitd** command.

Distributed System Management Interface Tool (DSMIT) Overview in the *Distributed SMIT 2.2 for*

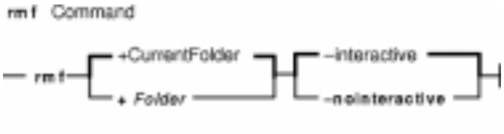
AIX: Guide and Reference.

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 Version 4.3 System User's Guide: Communications and Networks*.

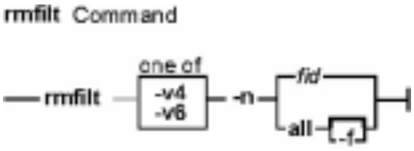
Peek, Jerry. *MH and xmh: E−mail for Users and Programmers.* Sebastopol, CA: O'Reilly & Associates, 1992.

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.

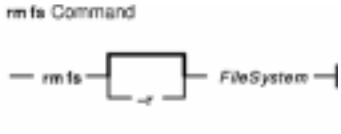
−v IP version of the filter rule you want to remove. Value **4** specifies IP version 4. Value **6** specifies IP version 6.

rmfs Command

Purpose

Removes a file system.

Syntax



rmfs [**−r**] *FileSystem*

Description

The **rmfs** command removes a file system. If the file system is a journaled file system (JFS), 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 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 Web−based System Manager File Systems application (**wsm fs** fast path) to run this command. 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.

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 Overview for System Management in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

Setting up and running Web−based System Manager in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

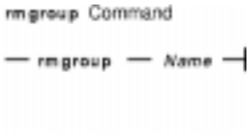
System Management Interface Tool (SMIT) Overview in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

rmgroup Command

Purpose

Removes a group.

Syntax



rmgroup*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.

You can use the Web−based System Manager Users application (**wsm users** fast path) to run this command. You could also use the System Management Interface Tool (SMIT) **smit rmgroup** fast path to run this command.

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

Examples

To remove the finance group, enter:

```
rmgroup finance
```

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.

Security Administration in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

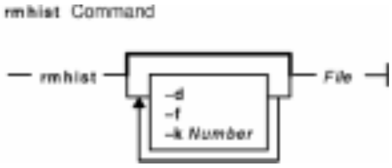
Setting up and running Web−based System Manager in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

rmhist Command

Purpose

Removes the history information from INed structured files.

Syntax



rmhist [**−d**] [**−f**] [**−k***Number*] *File* ...

Description

The **rmhist** command takes the specified file or files and removes the history information from each file. The files are backed up so that the old versions of the files are available as the *File***.bak** file. If any of the files are not structured files, a warning message is displayed, and no action is taken on these nonstructured files.

> **Note:** If the system date was set to a future date, the modification dates are invalid, and the **rmhist** command removes all history except the last history. It then resets the date for that record. You can correct this problem by using the **ghost** command. Enter the future date as the *mm/dd/yy* parameter to the **ghost** command, and then use the **rmhist** command to remove the history.

Flags

−d Deletes the **.bak** file after removing the history information.

−f Suppresses all messages about attempts to remove the history information from any files.

−k*Number* Keeps only the last specified number of days of history information. The default value of the *Number* variable is 1.

Examples

1. To remove history for all but the last three days of the `myfile` structured file and then delete the `myfile.bak` file, enter:

   ```
   rmhist −dk3 myfile
   ```

2. To keep only one day of history for the `testfile` and `afile` structured files, but not delete the `testfile.bak` and `afile.bak` files, enter:

   ```
   rmhist testfile afile
   ```

Related Information

The **e** command, **ghost** command, **readfile** command, **versions** command.

INed Editor Overview in *AIX Version 4.3 INed Editor User's Guide* introduces general concepts about the INed editor.

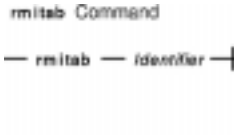
How to Access Previous Versions of a File with the INed Editor.

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

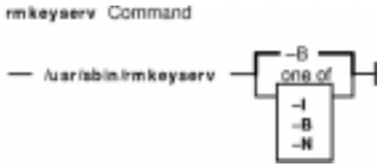
The **smit** command, **chitab** command, **lsitab** command, **mkitab** command, **init** command, **/etc/inittab** file.

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 Web−based System Manager File Systems application (**wsm fs** fast path) to run this command. 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 Version 4.3 System Management Concepts: Operating System and Devices*.

How to Start the NFS Daemons, How to Stop the NFS Daemons, How to Export a File System Using Secure NFS, How to Mount a File System Using Secure NFS in *AIX Version 4.3 System Management Concepts: Operating System and Devices*.

Setting up and running Web−based System Manager in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

System Management Interface Tool (SMIT) Overview in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

Network Information Service (NIS) in *AIX Version 4.3 Network Information Services (NIS and NIS+) Guide*.

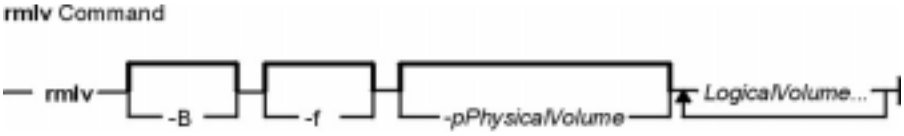
NIS Reference.

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.

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

You can use the Web−based System Manager Volumes application (**wsm lvm** fast path) to run this command. 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 *PhysicalVolum*e. 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 Version 4.3 System Management Guide: 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 (SMIT) Overview in *AIX Version 4.3 System Management Guide: Operating System and Devices* explains the structure, main menus, and tasks that are done with SMIT.

AIX HACMP/6000 Concepts and Facilities.

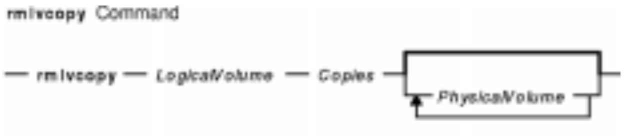
Setting up and running Web−based System Manager in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

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.

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

You can use the Web−based System Manager Volumes application (**wsm lvm** fast path) to run this command. You could also use the System Management Interface Tool (SMIT) **smit rmlvcopy** fast path to run this command.

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 Version 4.3 System Management Guide: Operating System and Devices* explains the Logical Volume Manager, physical volumes, logical volumes, volume groups, organization, ensuring data integrity, and allocation characteristics.

Setting up and running Web−based System Manager in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

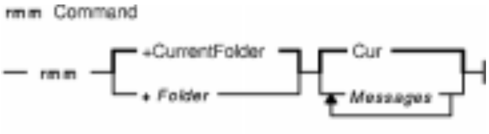
The System Management Interface Tool (SMIT) Overview in *AIX Version 4.3 System Management Guide: Operating System and Devices* explains the structure, main menus, and tasks that are done with SMIT.

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 Version 4.3 System User's Guide: Communications and Networks*.

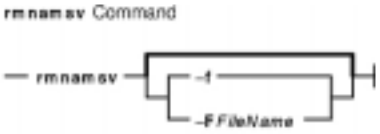
Peek, Jerry. *MH and xmh: E−mail for Users and Programmers.* Sebastopol, CA: O'Reilly & Associates, 1992.

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 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 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 Version 4.3 System Management Guide: Communications and Networks*.

Understanding the SMIT Interface for TCP/IP in *AIX Version 4.3 System Management Guide: Communications and Networks*.

Setting up and running Web−based System Manager in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

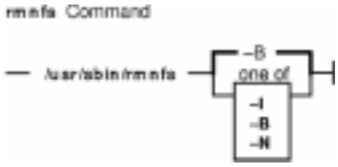
The **namerslv** command.

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 Version 4.3 System Management Guide: Communications and Networks*.

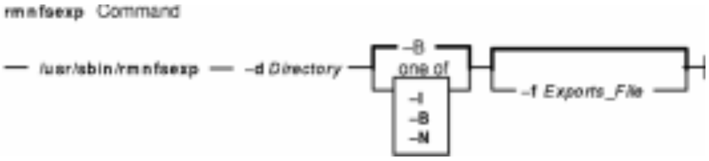
NFS Installation and Configuration in *AIX Version 4.3 System Management Guide: Communications and Networks*.

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, enter:

   ```
   rmnfsexp −d /usr −N
   ```

   In this example, the `/usr` directory is unexported immediately.

2. To unexport a directory immediately and after every system restart, enter:

   ```
   rmnfsexp −d /home/guest −B
   ```

3. To unexport a directory immediately from an exports file other than the **/etc/exports** file, enter:

   ```
   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.

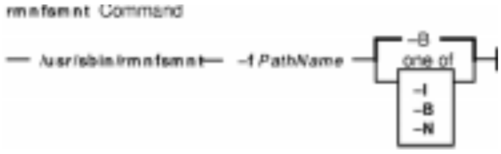
Network File System (NFS) Overview for System Management in *AIX Version 4.3 System Management Guide: Communications and Networks*.

rmnfsmnt Command

Purpose

Removes an NFS mount.

Syntax



/usr/sbin/rmnfsmnt −f *PathName* [**−I** | **−B** | **−N**]

Description

The **rmnfsmnt** command removes the appropriate entry from the **/etc/filesystems** file and unmounts the file system specified. When used with the **−N** flag, the **rmnfsmnt** command unmounts the file system and does not modify the **/etc/filesystems** file.

Flags

−B Removes the entry in the **/etc/filesystems** file and unmounts the directory. If no entry exists in the **/etc/filesystems** file, the flag makes no changes to the file. If the file system is not currently mounted, the flag does not attempt to unmount it. This flag is the default.

−f *PathName* Specifies the path name of the NFS−mounted file system.

−I Removes the entry specified by the path name from the **/etc/filesystems** file.

−N Unmounts the specified directory and does not modify the **/etc/filesystems** file.

Examples

1. To unmount a file system, enter:

   ```
   rmnfsmnt −f /usr/man −N
   ```

   In this example, the `/usr/man` file system is unmounted.

2. To remove a mount for a file, enter:

   ```
   rmnfsmnt −f /usr/local/man −B
   ```

In this example, the mount for the `/usr/local/man` file is removed.

File

/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 Version 4.3 System Management Guide: Communications and Networks*.

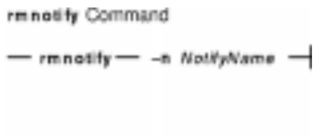
List of NFS Commands.

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 Version 4.3 System Management Guide: Operating System and Devices*.

System Resource Controller (SRC) Overview for Programmers in *AIX General Programming Concepts: Writing and Debugging Programs*.

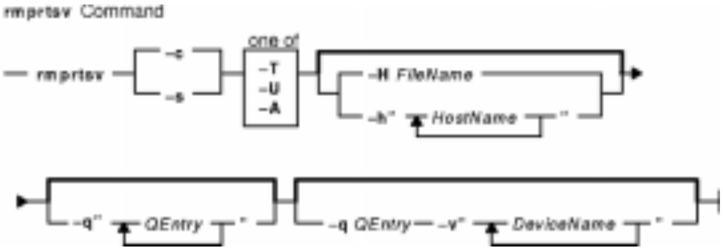
Understanding SRC Objects in *AIX General Programming Concepts: Writing and Debugging Programs*.

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.

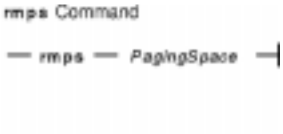
The SMIT Interface for TCP/IP in *AIX Version 4.3 System Management Guide: Communications and Networks*.

rmps Command

Purpose

Removes a paging space from the system along with any logical volume on which it resides.

Syntax



rmps*PagingSpace*

Description

The **rmps** command removes a paging space from the system along with any logical volume on which it resides. 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 Web−based System Manager File Systems application (**wsm fs** fast path) to run this command. You could also use the System Management Interface Tool (SMIT) **smit rmps** fast path to run this command.

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 Version 4.3 System Management Guide: Operating System and Devices* explains paging space and its allocation policies.

The File Systems Overview for System Management in *AIX Version 4.3 System Management Guide: Operating System and Devices* explains file system types, management, structure, and maintenance.

The Logical Volume Storage Overview in *AIX Version 4.3 System Management Guide: Operating System and Devices* explains the Logical Volume Manager, physical volumes, logical volumes, volume groups, organization, ensuring data integrity, and understanding the allocation characteristics.

Setting up and running Web−based System Manager in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

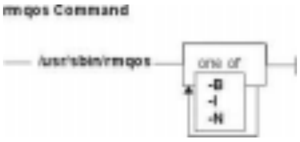
The System Management Interface Tool (SMIT) Overview in *AIX Version 4.3 System Management Guide: Operating System and Devices* explains the structure, main menus, and tasks that are done with SMIT.

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.

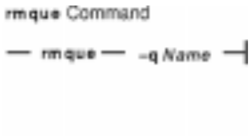
TCP/IP Quality of Service (QoS) in the *AIX Version 4.3 System Management Guide: Communications and Networks*.

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 Web−based System Manager Printer Queues application (**wsm printers** fast path) to run this command. 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 Version 4.3 Guide to Printers and Printing*.

Spooler Overview for System Management in *AIX Version 4.3 Guide to Printers and Printing*.

Removing a Print Queue in *AIX Version 4.3 Guide to Printers and Printing*.

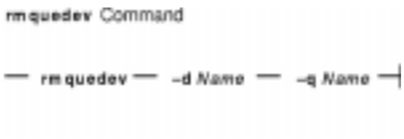
Setting up and running Web−based System Manager in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

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 Web−based System Manager Printer Queues application (**wsm printers** fast path) to run this command. 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 Version 4.3 Guide to Printers and Printing*.

Spooler Overview for System Management in the *AIX Version 4.3 Guide to Printers and Printing*.

Removing a Print Queue in the *AIX Version 4.3 Guide to Printers and Printing*.

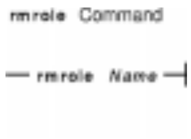
Setting up and running Web−based System Manager in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

rmrole Command

Purpose

Removes a role. This command applies only to AIX Version 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.

Security Administration in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

Administrative Roles Overview in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

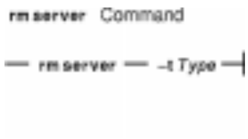
Setting up and running Web−based System Manager in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

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 Version 4.3 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 Version 4.3 System Management Guide: Operating System and Devices*.

System Resource Controller (SRC) Overview for Programmers in *AIX General Programming*

Concepts: Writing and Debugging Programs.

The **auditpr** command, **chserver** command, **mkserver** command, **startsrc** command, **stopsrc** command.

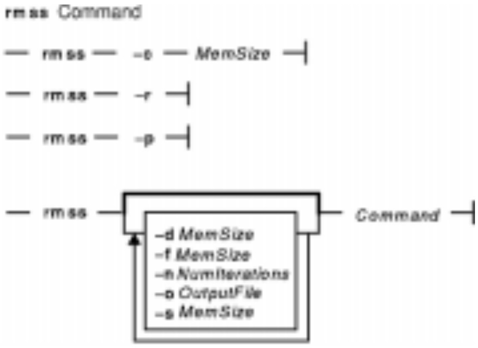
Auditing Overview in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

rmss Command

Purpose

Simulates a system with various sizes of memory for performance testing of applications.

Syntax



rmss −c*MemSize*

rmss −r

rmss −p

rmss [**−d***MemSize*] [**−f***MemSize*] [**−n***NumIterations*] [**−o** *OutputFile*] [**−s** *MemSize*] *Command*

Description

The **rmss** command simulates a system with various sizes of real memory, without having to extract and replace memory boards. By running an application at several memory sizes and collecting performance statistics, one can determine the memory needed to run an application with acceptable performance. The **rmss** command can be invoked for either of two purposes:

- To change the memory size and then exit, using the **−c** , **−p** , and **−r** flags. This allows the user to experiment freely with a given memory size.
- To function as a driver program, using the **−s** , **−f** , **−d** , **−n** , and **−o** flags. In this mode, the **rmss** command executes a specified command multiple times over a range of memory sizes, and displays important statistics describing command performance at each memory size. The command can be an executable or shell script file, with or without command line arguments.

The **−c**, **−p**, and **−r** flags are mutually exclusive. The **−c** flag changes the memory size; the **−p** flag displays the current memory size; and the **−r** flag resets the memory size to the real memory size of the machine.

The **−s**, **−f**, **−d**, **−n**, and **−o** flags are used in combination when the **rmss** command is invoked as a driver program to execute and measure the performance of a command (where a command is an executable or a shell script file) over a range of memory sizes. When invoked this way, the **rmss** command displays performance statistics, such as the response time of the command and the number of page−ins that occurred while the command ran, for each memory size. These statistics, which are also written to a file, are described in example 7 .

The **−s** and **−f** flags specify the starting and ending points of the range, while the **−d** flag specifies the increment between memory sizes within the range. The **−n** flag is used to specify the number of times to run the command at each memory size, and the **−o** flag is used to specify the name of an output file into which to write the **rmss** report. The *Command* parameter specifies the command to be run and measured at each memory size.

> **Note:** The **rmss** command reports "usable" real memory. On machines where there is bad memory or where the system is using the memory, **rmss** reports the amount of real memory as the amount of physical real memory minus the memory that is bad or in use by the system. For example, using the **rmss−r** flag might report:

```
Simulated Memory Size changed to 79.9062MB
```

This could be a result of some pages being marked bad or a result of a device that is reserving some pages for its own use (and thus not available to the user).

> **Note:** The **rmss** command may underestimate the number of page−ins that are required to run an application if the application, combined with background processes such as daemons, accesses a lot of different files (including directory files). The number of different files that must be accessed in order to cause such results is approximately 250 files per 8MB of simulated memory size. The following table gives the approximate number of different files that, when accessed at the given simulated memory size, may result in the **rmss** command underestimating page−in requirements.

Simulated Memory Size (MB)	**Access to Different Files**
8	250
16	500
24	750
32	1000
48	1500
64	2000
128	4000
256	8000

You can use the **filemon** command to determine the number of files accessed while your command runs, if you suspect that it may be accessing many different files.

Flags

−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 Display the current simulated memory size.

−r Reset 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. Here, 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

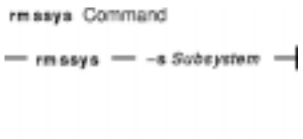
The **filemon** command.

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 Version 4.3 System Management Guide: Operating System and Devices* 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 Version 4.3 System Management Guide: Operating System and Devices*.

System Resource Controller (SRC) Overview for Programmers in *AIX General Programming Concepts: Writing and Debugging Programs*.

Auditing Overview in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

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.

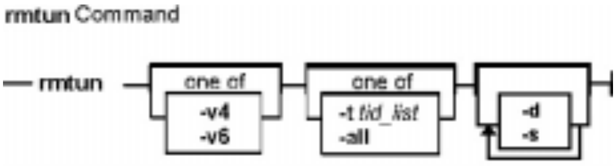
The **rmt** special file.

rmtun Command

Purpose

Deactivates operational tunnel(s) and optionally removes tunnel definition(s).

Syntax



rmtun−v4|**6−t***tid_list* | **all** [**−d**] [**−s**]

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.

−s Shuts down the IBM session key engine of IP version 4. This action deactivates all IBM tunnels. This flag may not be specified with any other flags.

−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

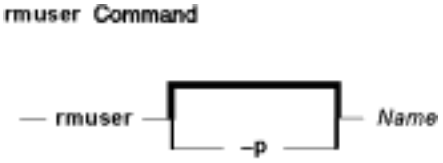
The **chtun** command, **exptun** command, **gentun** command, **imptun** command, **lstun** command, and **mktun** command.

rmuser Command

Purpose

Removes a user account.

Syntax



rmuser [**−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.

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 Web−based System Manager Users application (**wsm users** fast path) to run this command. 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.

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, enter:

   ```
   rmuser −p 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 Security Administration in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

For more information about administrative roles, refer to Administrative Roles Overview in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

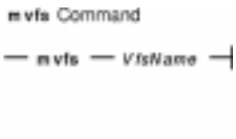
Setting up and running Web−based System Manager in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

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 Overview for System Management in *AIX Version 4.3 System Management Guide: Operating System and Devices* explains file system types, management, structure, and maintenance.

rmvirprt Command

Purpose

Removes a virtual printer.

Syntax

rmvirprt Command
rmvirprt
-qPrinterQueueName
-dQueueDeviceName

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 Web−based System Manager Printer Queues application (**wsm printers** fast path) to run this command. You could 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, enter:

```
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 **mkvirprt** command, **lsvirprt** command, **chvirprt** command, **smit** command.

Printer Overview for System Management in the *AIX Version 4.3 Guide to Printers and Printing*.

Spooler Overview for System Management in the *AIX Version 4.3 Guide to Printers and Printing*.

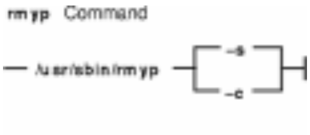
Setting up and running Web−based System Manager in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

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 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 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 Version 4.3 System Management Concepts: Operating System and Devices*.

Setting up and running Web−based System Manager in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

System Management Interface Tool (SMIT) Overview in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

Network Information Service (NIS) in *AIX Version 4.3 Network Information Services (NIS and NIS+) Guide*.

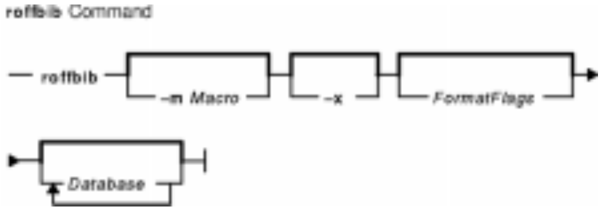
NIS Reference.

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

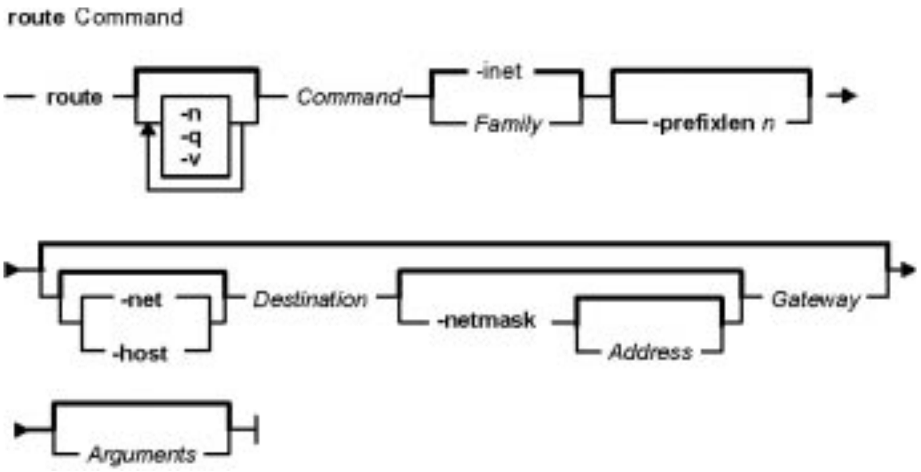
The **addbib** command, **indxbib** command, **lookbib** command, **nroff** command, **refer** command, **sortbib** command, **troff** command.

route Command

Purpose

Manually manipulates the routing tables.

Syntax



route [**−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

−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.

−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 Version 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 Version 4.2.1 or later.

−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.

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 lockup 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 Version 4.3 System Management Guide: Communications and Networks*.

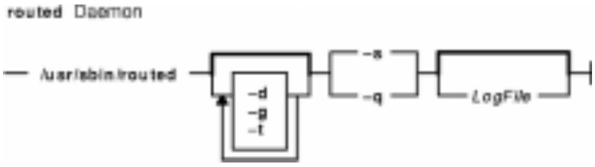
Xerox Network Systems (XNS) Overview for Programming in *AIX Version 4.3 Communications Programming Concepts*.

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, enter:

   ```
   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 normally, enter the following:

   ```
   stopsrc −s routed
   ```

   This command stops the daemon.

3. To get a short−status report from the **routed** daemon, enter 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, enter 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`, enter 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.

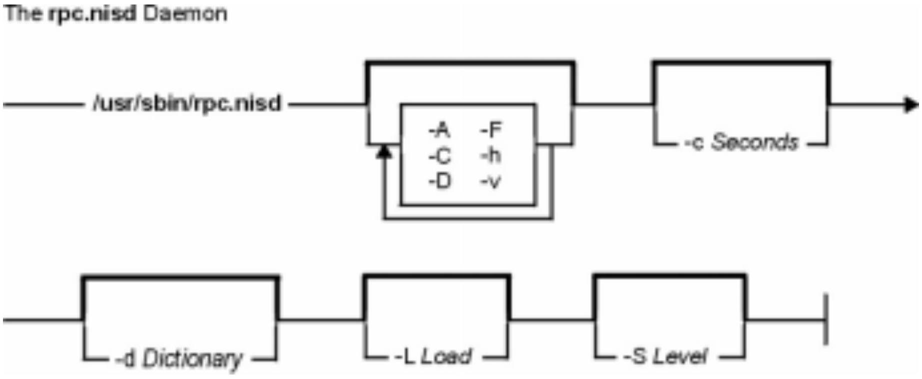
TCP/IP Routing, TCP/IP Daemons, TCP/IP Protocols in *AIX Version 4.3 System Management Concepts: Operating System and Devices*.

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

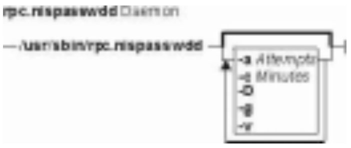
The **nis_cachemgr** daemon, the **nisinit** command, the **nissetup** command, and the **rpc.nispasswdd** daemon.

rpc.nispasswd Daemon

Purpose

NIS+ password update daemon.

Syntax



/usr/sbin/rpc.nispasswd [[**−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** deamon 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.

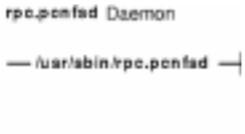
The **rpc.nisd** daemon, **syslogd** daemon.

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. Once 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. Since 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 Version 4.3 System Management Guide: Communications and Networks*.

Printer Overview for System Management in the *AIX Version 4.3 Guide to Printers and Printing*.

List of NFS Commands in *AIX Version 4.3 System Management Guide: Communications and Networks*.

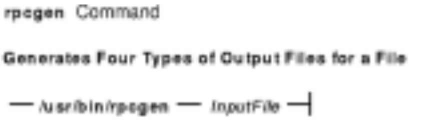
rpcgen Command

Purpose

Generates C code to implement an RPC protocol.

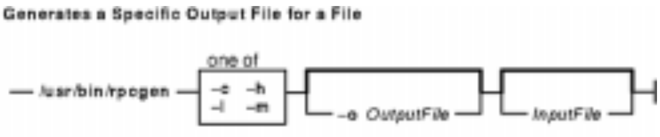
Syntax

To Generate Four Types of Output Files for a File



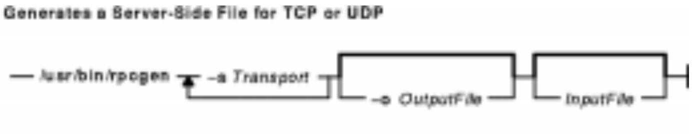
/usr/bin/rpcgen*InputFile*

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.

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 Version 4.3 System Management Guide: Communications and Networks*.

Remote Procedure Call (RPC) Overview for Programming in *AIX Communications Programming Concepts*.

List of NFS Commands.

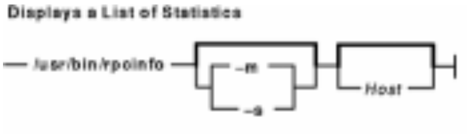
rpcinfo Command

Purpose

Reports the status of Remote Procedure Call (RPC) servers.

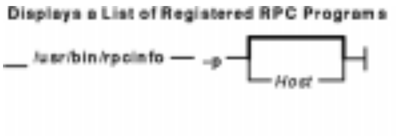
Syntax

To Display a List of Statistics



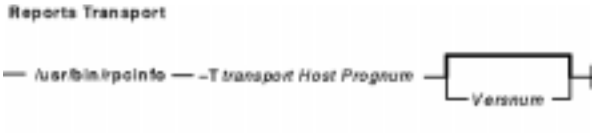
/usr/bin/rpcinfo [**−m** | **−s**] [*Host*]

To Display a List of Registered RPC Programs



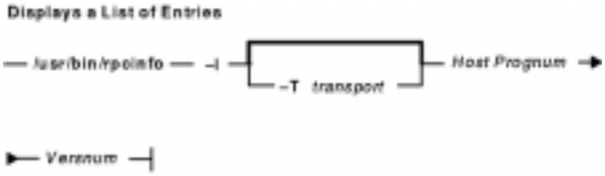
/usr/bin/rpcinfo −p [*Host*]

To Report Transport



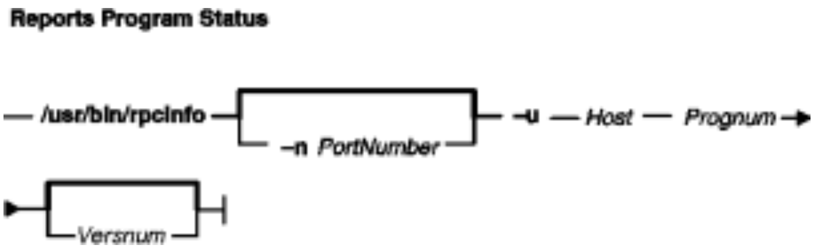
/usr/bin/rpcinfo −T*transport Host Prognum* [*Versnum*]

To Display a List of Entries



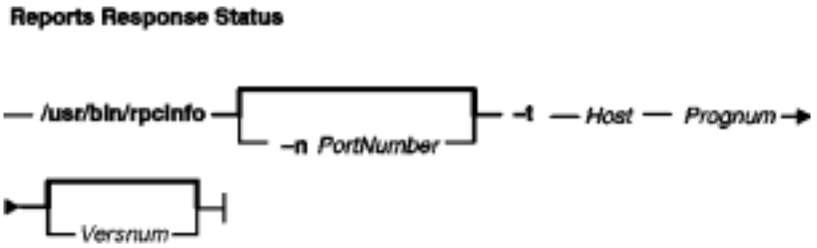
/usr/bin/rpcinfo −l [**−T** *transport*] *Host Prognum Versnum*

To Report Program Status



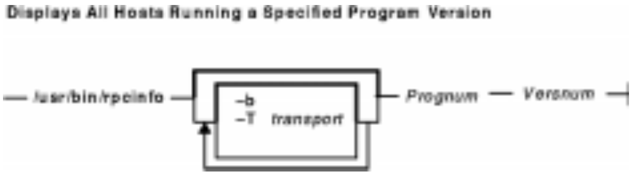
/usr/bin/rpcinfo [**−n***PortNum*] **−u** *Host Prognum* [*Versnum*]

To Report Response Status



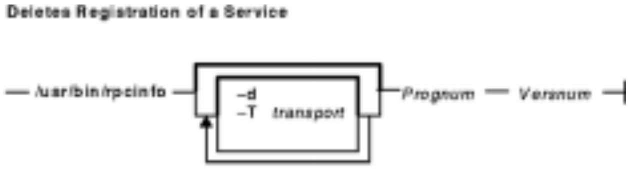
/usr/bin/rpcinfo [**−n***PortNum*] **−t** *Host Prognum* [*Versnum*]

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 Version 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 Version 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 Version 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 Version 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 Version 4.3 System Management Guide: Communications and Networks*.

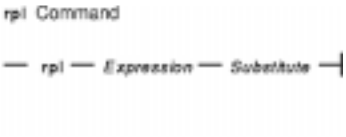
List of NFS Commands.

rpl Command

Purpose

Replaces all occurrences of a string in a file.

Syntax



rpl*ExpressionSubstitute*

Description

The **rpl** command reads standard input, replaces all occurrences of the first parameter, *Expression*, with the second parameter, *Substitute*, and then writes the result to standard output.

You can use the **rpl** command as a filter with the **Do** function (the Alt−X key sequence), which runs the filter, or as a filter command from the system prompt. For information about keyboard layouts, press the F1 key (the **Help** function) within the INed editor. If you enter the **rpl** command from the system prompt, the command reads text from standard input and writes it to standard output.

If you enter the **rpl** command with a file name from the system prompt, it reads the text from that file, replaces all text, and writes the result to standard output.

The first value is usually a simple text string. You can also use characters (called pattern−matching characters) to form patterns that match classes of text strings. The characters that have special meanings in forming patterns are:

^ At the beginning of a pattern, a ^ (circumflex) matches the empty string at the beginning of a line.

. A . (period) matches any character except a new−line character.

$ A $ (dollar sign) matches the end of a line.

\*Number* Matches the character whose ASCII code is given by the *Number* variable, whose value can be up to three octal digits, except the octal digits 012 (new−line character). Because the **rpl** command works on only one line at a time, it is impossible to match a new−line character.

[ ] Matches any character in the string of characters enclosed in [ ] (brackets), but no others. If the first character of the string is a ^ (circumflex), the pattern matches any character except the new−line character and the characters in the string.

− Matches any character in a range. Within brackets, the − (minus) specifies a range. For example, `[a−d]` is equivalent to `[abcd]`. The − (minus) stands for itself if you use it as the first or last character. For example, this pattern `[ ] − ]` matches the characters] and −.

***** A pattern followed by an * (asterisk) means zero or more times.

+ A pattern followed by a + (plus) means one or more times. For example, `[0−9]+` is equivalent to `[0−9][0−9]*`.

{*Minimum*,*Maximum***}** Integer values enclosed in { } (braces) indicate the number of times the preceding regular expression is applied. The *Minimum* number is followed by the *Maximum* number, which must be less than 256. Specifying only the minimum number {*Minimum*}, gives the exact number of times the regular expression is applied. Specifying {*Minimum*,} is the same as {*Minimum*,*Infinity*}. The + (plus) and * (asterisk) are equivalent to {1,} and {0,}, respectively.

(...) Parentheses are used for grouping. An operator (for example, *, +, { }) can work on a single character or a regular expression enclosed in parentheses; for example, `(a*(cb+)*)$`.

All of the preceding characters are special. If you want to use them as themselves, you must precede each with a **** (reverse slash). Also, to avoid conflicts with AIX special characters, enclose the *Expression* and *Substitute* parameters in single quotation marks when they contain any special characters.

To delete a string using the **rpl** command, use the null string `""` (two double quotation marks) as the replacement string.

If lines are longer than 256 characters, if you do not give two values, or if the first value is the null string, the **rpl** command exits with the value −2.

Examples

See the following for examples of replacing character strings.

> **Note:** If you are using a locale other than **En_US**, the following examples may not include all alphabetic characters.

1. To replace a legal FORTRAN identifier (an alphabetic character followed by as many as five alphanumeric characters) with `xxx`, enter:

   ```
   rpl '[A−Za−z][A−Za−z0−9]{0,5}' xxx
   ```

2. To replace all occurrences of the character string `sky` with `bluesky` in the `skyfile` file and put the result in the `bluefile` file, enter:

   ```
   rpl sky "blue sky" <skyfile >bluefile
   ```

Related Information

The **e** command.

Editor Overview in *AIX Version 4.3 INed Editor User's Guide* introduces general concepts about the INed editor.

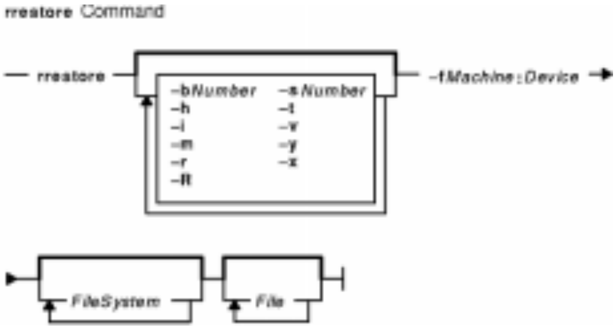
Running AIX and Filter Commands with the INed Editor in *AIX Version 4.3 INed Editor User's Guide*.

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 Version 4.3 System Management Guide: Operating System and Devices*.

File Systems Overview for System Management in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

Directory Overview in *AIX Version 4.3 System User's Guide: Operating System and Devices*.

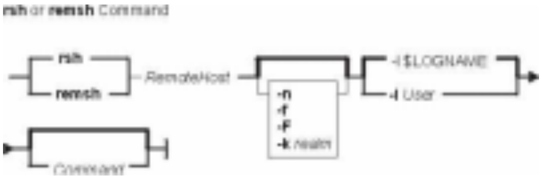
Files Overview in *AIX Version 4.3 System User's Guide: Operating System and Devices*.

rsh or remsh Command

Purpose

Executes the specified command at the remote host or logs into the remote host.

Syntax



{ **rsh** | **remsh** } *RemoteHost* [**−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:** Since 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 AIX 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 non−AIX 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. Normally, 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 in to the remote host. Again, for successful login, the local user must have a user account on the remote host.

Flags

−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.

−fCauses 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.

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.

Related Information

The **ftp** command, **rcp** command, **rexec** command, **rlogin** command, **telnet**, **tn**, or **tn3270** command.

The **rshd** and **krshd** daemon.

The **kvalid_user** function.

The **hosts.equiv** file format, **.rhosts** file format.

Network Overview in *AIX Version 4.3 System Management Guide: Communications and Networks*.

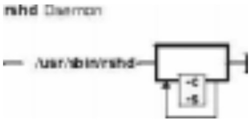
Secure Rcmds in *AIX Version 4.3 System User's Guide: Communications and Networks*.

rshd Daemon

Purpose

Provides the server function for remote command execution.

Syntax



> **Note:** The **rshd** daemon is normally 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** file 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. Once **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. Entering `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, enter the following:

    ```
    startsrc −t shell
    ```

    This command starts the **rshd** subserver.

2. To stop the **rshd** daemon normally, enter 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, enter 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, enter 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 Version 4.3 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 Version 4.3 System Management Guide: Communications and Networks*.

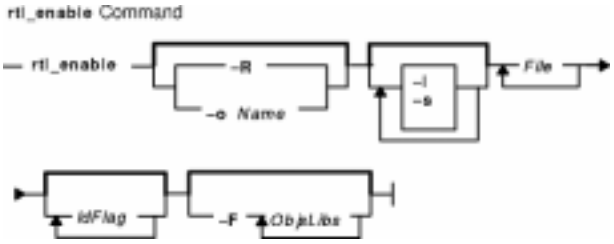
List of NFS Commands.

rtl_enable Command

Purpose

Relinks shared objects to enable the runtime linker to use them. This command only applies to AIX Version 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, overridding 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 Version 4.3 General Programming Concepts: Writing and Debugging Programs*.

runacct Command

Purpose

Runs daily accounting.

Syntax

runacct Command
/usr/sbin/acct/runacct
mmdd
State

/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.

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 Version 4.3 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 Version 4.3 System Management Guide: Operating System and Devices*.

Setting Up an Accounting System in *AIX Version 4.3 System Management Guide: Operating System and Devices* describes the steps you must take to establish an accounting system.

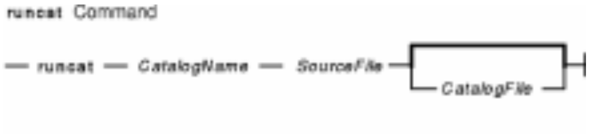
See the Accounting Commands in *AIX Version 4.3 System Management Guide: Operating System and Devices* for a list of accounting commands that can be run automatically or entered from the keyboard.

runcat Command

Purpose

Pipes output data from the **mkcatdefs** command to the **gencat** command.

Syntax



runcat*CatalogNameSourceFile* [*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.

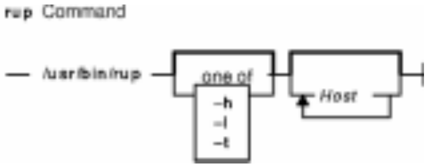
Message Facility Overview for System Management in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

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.

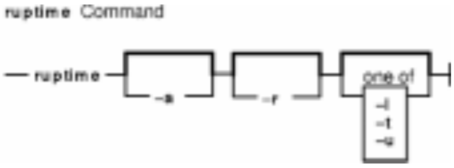
Network File System (NFS) Overview for System Management in *AIX Version 4.3 System Management Guide: Communications and Networks*.

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 Version 4.3 System Management Guide: Communications and Networks*.

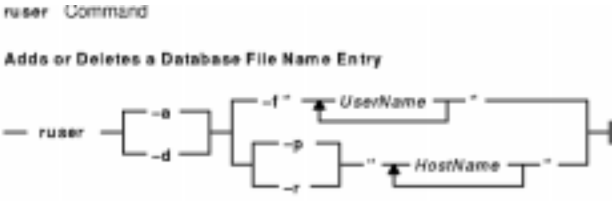
ruser Command

Purpose

Directly manipulates entries in three separate system databases that control foreign host access to programs.

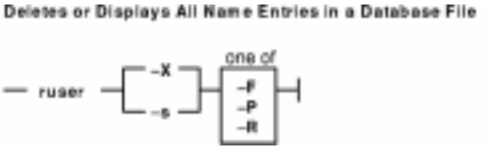
Syntax

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 a Web−based System Manager application (**wsm users** fast path) to run this command. You could also use the System Management Interface Tool (SMIT) **smit users** fast path to run this command.

OR

```
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, enter 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 delete as well the same name from the database that controls remote access for the **rlogin**, **rcp**, and **rsh** commands (**/etc/hosts.equiv**), enter:

   ```
   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.

Setting up and running Web−based System Manager in *AIX Version 4.3 System Management Guide: Operating System and Devices*.

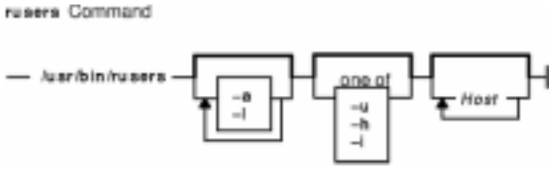
Understanding the TCP/IP Reference in *AIX Version 4.3 System Management Guide: Communications and Networks*.

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 **rup** 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 Version 4.3 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 Version 4.3 System Management Guide: Communications and Networks*.

List of NFS Commands.

rwall Command

Purpose

Sends messages to all users on the network.

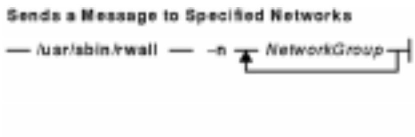
Syntax

To Send a Message to Specified Hosts

rwall Command
Sends a Message to Specified Hosts
/usr/sbin/rwall
HostName

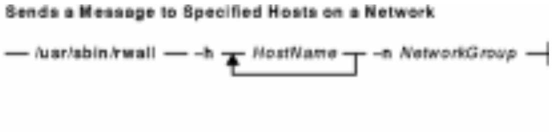
/usr/sbin/rwall*HostName* ...

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 Version 4.3 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 Version 4.3 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.

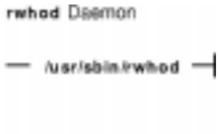
Network Overview in *AIX Version 4.3 System Management Guide: Communications and Networks*.

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 Version 4.3 System Management Concepts: Operating System and Devices*.

Vos remarques sur ce document / Technical publication remark form

Titre / Title : Bull AIX Commands Reference Vol.4 named to rwhod

Nº Reférence / Reference Nº : 86 A2 41JX 02

Daté / Dated : April 2000

ERREURS DETECTEES / ERRORS IN PUBLICATION

AMELIORATIONS SUGGEREES / SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION

Vos remarques et suggestions seront examinées attentivement.
Si vous désirez une réponse écrite, veuillez indiquer ci-après votre adresse postale complète.

Your comments will be promptly investigated by qualified technical personnel and action will be taken as required.
If you require a written reply, please furnish your complete mailing address below.

NOM / NAME : ______________________ Date : __________

SOCIETE / COMPANY : ______________________

ADRESSE / ADDRESS : ______________________

____

Remettez cet imprimé à un responsable BULL ou envoyez-le directement à :

Please give this technical publication remark form to your BULL representative or mail to:

BULL ELECTRONICS ANGERS
CEDOC
34 Rue du Nid de Pie – BP 428
49004 ANGERS CEDEX 01
FRANCE

Technical Publications Ordering Form

Bon de Commande de Documents Techniques

To order additional publications, please fill up a copy of this form and send it via mail to:
Pour commander des documents techniques, remplissez une copie de ce formulaire et envoyez-la à :

BULL ELECTRONICS ANGERS
CEDOC
ATTN / MME DUMOULIN
34 Rue du Nid de Pie – BP 428
49004 ANGERS CEDEX 01
FRANCE

Managers / Gestionnaires :
Mrs. / Mme : **C. DUMOULIN** +33 (0) 2 41 73 76 65
Mr. / M : **L. CHERUBIN** +33 (0) 2 41 73 63 96

FAX : +33 (0) 2 41 73 60 19
E–Mail / Courrier Electronique : srv.Cedoc@franp.bull.fr

Or visit our web site at: / Ou visitez notre site web à:
`http://www–frec.bull.com` (PUBLICATIONS, Technical Literature, Ordering Form)

CEDOC Reference # Nº Référence CEDOC	**Qty** Qté	**CEDOC Reference #** Nº Référence CEDOC	**Qty** Qté	**CEDOC Reference #** Nº Référence CEDOC	**Qty** Qté
__ __ ____ _ [__]		__ __ ____ _ [__]		__ __ ____ _ [__]	
__ __ ____ _ [__]		__ __ ____ _ [__]		__ __ ____ _ [__]	
__ __ ____ _ [__]		__ __ ____ _ [__]		__ __ ____ _ [__]	
__ __ ____ _ [__]		__ __ ____ _ [__]		__ __ ____ _ [__]	
__ __ ____ _ [__]		__ __ ____ _ [__]		__ __ ____ _ [__]	
__ __ ____ _ [__]		__ __ ____ _ [__]		__ __ ____ _ [__]	
__ __ ____ _ [__]		__ __ ____ _ [__]		__ __ ____ _ [__]	
[_ _] : **no revision number means latest revision** / pas de numéro de révision signifie révision la plus récente					

NOM / NAME : ______________________________ Date : __________

SOCIETE / COMPANY : ______________________________

ADRESSE / ADDRESS : ______________________________

____

PHONE / TELEPHONE : ____________________ FAX : ____________________

E–MAIL : ______________________________

For Bull Subsidiaries / Pour les Filiales Bull :
Identification: ______________________________

For Bull Affiliated Customers / Pour les Clients Affiliés Bull :
Customer Code / Code Client : ______________________________

For Bull Internal Customers / Pour les Clients Internes Bull :
Budgetary Section / Section Budgétaire : ______________________________

For Others / Pour les Autres :
Please ask your Bull representative. / Merci de demander à votre contact Bull.

PLACE BAR CODE IN LOWER
LEFT CORNER

BULL ELECTRONICS ANGERS
CEDOC
34 Rue du Nid de Pie – BP 428
49004 ANGERS CEDEX 01
FRANCE

ORDER REFERENCE
86 A2 41JX 02

Bull

Utiliser les marques de découpe pour obtenir les étiquettes.
Use the cut marks to get the labels.

AIX

AIX Commands
Reference Vol.4
named to rwhod

86 A2 41JX 02

AIX

AIX Commands
Reference Vol.4
named to rwhod

86 A2 41JX 02

AIX

AIX Commands
Reference Vol.4
named to rwhod

86 A2 41JX 02