



# Bull

## Quick Reference Guide

AIX

L



# Bull

## Quick Reference Guide

AIX

---

**Software**

June 1998

**Bull Electronics Europe S.A.**  
Service CEDOC  
331 Avenue Patton - BP 428  
49004 ANGERS CEDEX 01  
FRANCE

ORDER REFERENCE  
**86 A2 55AP 02**



The following copyright notice protect this book under the Copyright laws of the United States and other countries which prohibit such actions as, but not limited to, copying, distributing, modifying, and making derivative works.

Copyright © Bull S.A. 1992, 1998

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.

### **Trademarks and Acknowledgements**

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

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

UNIX is a registered trademark in the USA and other countries licensed exclusively through X/Open.

*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.*



---

# Contents

<b>Introduction</b> .....	<b>v</b>
<b>Operating System Tasks</b> .....	<b>1</b>
Working with Files and Directories ....	1
Managing File and Directory Access and Ownership .....	8
Editing Files .....	11
Printing .....	12
Working with Processes and Commands .....	16
Working with Shells .....	20
Using Web-based System Manager on AIX Version 4.3 or Later .....	21
Using SMIT .....	21
Getting Information and Help .....	22
Backing Up and Restoring .....	23
Analyzing Space Utilization .....	28
Solving Problems .....	28
<b>Using Command Line Key Functions</b> ..	<b>31</b>
<b>Using the vi Editor</b> .....	<b>32</b>
<b>Using the ed Editor</b> .....	<b>36</b>
<b>Working with the Korn Shell</b> .....	<b>38</b>
<b>System Management Tasks Using Web-based System Manager on AIX Version 4.3 or Later</b> .....	<b>50</b>
<b>Using InfoExplorer Key Functions on AIX Versions 4.1 and 4.2</b> .....	<b>53</b>
<b>Index</b> .....	<b>57</b>

L



---

## Introduction

This guide is intended for a wide range of users performing common tasks on their systems. It is intended to cover *only* common tasks and therefore does not contain complete information.

## Highlighting

The following highlighting conventions are used in this guide:

**Bold** Identifies command words, keywords, files, directories, and other items with names predefined by the system.

*Italics* Identifies parameters whose actual names or values are 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.

## Reading Syntax Diagrams

Usually, a command follows this syntax:

### Command Flags or *Parameters Objects*

[ ] Material within brackets is optional.

{ } Material within braces is required.

| Indicates an alternative. Only one of the options can be chosen.

... Indicates that one or more of the kinds of parameters or objects preceding the ellipsis can be entered.



## **ISO 9000**

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





## Related Publications

To understand the full range of a command's capabilities and restrictions, refer to the HTML-based AIX documentation (Version 4.3), InfoExplorer (Versions 4.1 and 4.2), or to the hardcopy publications listed below:

### Version 4.3

- *AIX Version 4.3 Commands Reference*, Order Number 86 A2 38JX to 43JX
- *AIX Version 4.3 Quick Beginnings*, Order Number 86 A2 75HX
- *AIX Version 4.3 System User's Guide: Operating System and Devices*, Order Number 86 A2 97HX
- *AIX Version 4.3 System User's Guide: Communications and Networks*, Order Number 86 A2 98HX
- *AIX Version 4.3 Guide to Printers and Printing*, Order Number 86 A2 37JX
- *AIX Version 4.3 System Management Guide: Operating System and Devices*, Order Number 86 A2 99HX

### Versions 4.1 and 4.2

- *AIX Version 4 Commands Reference*, Order Number 86 A2 73AP to 78AP
- *AIX Version 4 Getting Started*, Order Number 86 A2 50AP
- *AIX Version 4 System User's Guide: Operating System and Devices*, Order Number 86 A2 51AP
- *AIX Version 4 System User's Guide: Communications and Networks*, Order Number 86 A2 52AP
- *AIX Version 4 Guide to Printers and Printing*, Order Number 86 A2 83AT
- *AIX Version 4 System Management Guide: Operating System and Devices*, Order Number 86 A2 53 AP

L



---

# Operating System Tasks

## Working with Files and Directories

### Displaying Files

**cat** [ *File...* ]

Displays or concatenates files.

**Note:** Do not redirect output to one of the input files using the > redirection symbol. This will truncate the input file.

```
cat file1
```

Displays `file1` on your workstation.

```
cat file1 file2 ... filen > bigfile
```

Concatenates `file1` through `filen` into `bigfile`.

```
cat file1 >> file2
```

Appends `file1` to the end of `file2`.

```
cat >> file1  
added text
```

**Ctrl-D**

Appends the string “`added text`” to the end of `file1`. Press the Ctrl-D sequence to indicate you are finished.

**pg** [ *File...* ]

Displays *File*, one page at a time.



## Changing Current Directory

**cd** [ *Directory* ]

Changes the current directory.

```
cd
```

Changes to **\$HOME** directory.

```
cd ..
```

Changes to parent directory of current directory.

```
cd subdirectory1
```

Changes to `subdirectory1` of current directory.

## Copying Files or Directories

**cp** [-R ] *Source Target*

Copies files or directories.

**-R** Causes the special files to be recreated in the *Target* directory.

```
cp myfile.c prog.c
```

Copies `myfile.c` to `prog.c`. If the `prog.c` file does not already exist, **cp** command creates it.

```
cp petunia /home/garden/flowers
```

Copies the `petunia` file to the directory `/home/garden/flowers`.

```
cp -R /home/garden/flowers  
/home/garden/plants
```

Copies the `flowers` directory, including all its files, subdirectories, and the files in those subdirectories, to the `plants` directory. The result is `/home/garden/plants/flowers`.

## Deleting Files

**del** *File...*

Deletes *File* after asking for confirmation.



The `del` command ignores file protection, allowing the owner of a file to delete a write-protected file. `<Enter>` or `<y>` affirms that you want the specified files deleted.

```
del *.bak
```

Deletes all files ending with `.bak`.

**rm** [ **-f -i -r** ] { *File...* | *Directory...*}

Deletes (unlinks) files or directories.

- f** Does not prompt you before deleting a write-protected file.
- i** Prompts you before deleting each file.
- r** Removes directories and their contents.

```
rm file1
```

Deletes `file1`. If `file1` has two or more links (names), then the named file is removed but its contents remain available through its other links.

```
rm -i dir1/*
```

Displays file names in `dir1` and prompts you for permission to delete.

```
rm -i -r dir1
```

Deletes contents of subdirectories of `dir1`, then removes `dir1` itself, after prompting you for permission to delete.

## Comparing Text Files

**diff** [ **-w -i -e** ] *File1 File2*

Compares text files and tells you what lines must be changed to make the files agree.

- w** Ignores tabs and spaces.
- i** Ignores case of letters.
- e** Produces output usable by the **ed** editor to convert *File1* to *File2*.



```
diff -e file1 file2 > change1.to2
```

Creates a file `change1.to2` containing commands that change `file1` to `file2`.

**diff** [ **-r -s** ] *Directory1 Directory2*

Sorts the contents of directories by name and performs the regular **diff** command on text files that are different.

**-r** Compares files in common subdirectories.

**-s** Reports identical files.

## Finding Files

**find** *Path...* [ *Expression* ]

Finds files using a matching expression.

**-name** *Filename*  
Finds file called *Filename*.

**-inum** *Number*  
Finds files with index node *Number*.

**-print** Displays current path name.

```
find / -name .file1
```

Finds and writes complete path names of all files in **/(root)** and its subdirectories named `.file1`.

## Transferring Files

**ftp** [ **-g -i -v** ] [ *HostName* ]

Transfers files between a local and a remote host.

This starts an interactive process, beginning with logging in on the remote host. A `>` is a subprompt for this command, asking for more input.

**-g** Does not expand file names.

**-i** Turns off interactive prompting in multiple file transfers.



- v** Displays responses from remote server and provides data transfer statistics.

## Searching Files for Patterns

**grep** [ **-c -i -v -w** ] *Pattern...* [ *File...* ]

Searches a file for a pattern.

- c** Displays only a count of matching lines.
- i** Ignores case of letters when making comparisons.
- v** Displays all lines that do not match the pattern.
- w** Does a word search.

*Pattern* A limited regular expression such as a character string or a range of characters (for instance, [ a-z ] ). A range of characters is usually enclosed in quotation marks.

**^** The caret preceding a range of characters indicates that **grep** is to search for a string beginning with any of the included characters.

```
grep "[a-zA-Z]" file1
```

Displays all lines in `file1` that begin with a letter.

```
grep -v "#" file1
```

Displays all lines in `file1` not beginning with a # character.

## Displaying Lines of a File

**head** [ **-Count** ] [ *File...* ]

Displays the first few lines of a file.

- Count** Specifies the number of lines to be displayed (default is 10).



```
head -5 myfile
```

Displays the first five lines of the file `myfile`.

## Linking Files

**ln** [-s] { *SourceFile* [ *TargetFile* ] | *SourceFile...*  
*TargetDirectory* }

Links files.

**-s** Creates a symbolic link, containing the name of the file to which it is linked.

```
ln file1 file2
```

Gives the name `file2` to `file1`. If `file2` does not already exist, it is created. If `file2` does exist, its contents are replaced by the contents of `file1`.

```
ln file1 directory2
```

Links `file1` to `directory2/file1`, the same file name in a different directory.

```
ln dir1/* .
```

Links all files in `dir1` to the `(.)` current directory.

## Listing Directory Contents

**ls** [-a -d -l] [ *File...* ] [ *Directory...* ]

Lists directory contents.

**-a** Lists all directory contents including `(.)` hidden files.

**-d** Lists information about a directory, not its files.

**-l** Lists directory contents in long form.

```
ls -a
```

Lists all files in the current directory.

```
ls -l myfile .profile
```





Displays long listing with detailed information about `myfile` and `.profile`.

## Creating a Directory

**mkdir** *Directory...*

Creates a directory.

## Moving or Renaming Files

**mv** [**-i** | **-f**] {{ *File...* | *Directory...* } *Directory* | { *File* | *Directory* } *NewName* }

Moves or renames files and directories.

The **mv** command overwrites files. Using the **-i** flag causes a prompt if an overwrite is about to occur.

- i** Prompts you before moving a file or directory over an existing one. The move continues if you answer with a `<y>`.
- f** Does not prompt you before overwriting an existing file.

```
mv file1 file2
```

Renames `file1` to `file2`.

```
mv Directory1 Directory2
```

Makes `Directory1` a subdirectory of `Directory2`, if `Directory2` exists. Otherwise, renames `Directory1` to `Directory2`.

```
mv file1 Directory1/file2
```

Moves `file1` to `Directory1`, renaming it to `file2`.

## Displaying Directory Path Name

**pwd**

Displays path name of your current directory.



## Removing a Directory

### **rmdir** *Directory...*

Removes a directory.

```
rm dir1/* dir1/.*;rmdir dir1
```

Removes contents of `dir1`, then removes `dir1`. The command line `rm -i -r dir1` is equivalent in function.

## Writing a File to Standard Output

### **tail** [ **-f -r** ] [ **-n** *Number* ] [ *File* ]

Writes a file to standard output, beginning at a specified point.

**-f** Continues to read and copy additions to input file as they become available.

**-r** Displays lines from the end in reverse order.

**-n** *Number*

Begins reading file at line *Number*.  
The default is 20.

```
tail -n 50 myfile
```

Starts reading file `myfile` 50 lines from the end of the file.

## Managing File and Directory Access and Ownership

### Changing File or Directory Ownership

#### **chgrp** *Group* { *File...* | *Directory* }

Changes group ownership of a file or directory to the specified *Group*.



```
chgrp staff myfile
```

Changes the group ownership of the file `myfile` to the group `staff`.

**chown** [ **-R** ] *Owner[:Group]* { *File...* | *Directory...* }

Changes user (owner and, if specified, group) associated with a file or directory. The user specified is the owner after the change. Only the root user can change the owner of a file. You can change the group of a file only if you are the root user or if you own the file.

**-R** Includes all subdirectories and files.

*:Group* Changes group ownership of files as well as owner. Do not put a space between *Owner* and *:Group*.

## Changing Permissions

**chmod** [ **-R** ] { [**ugo**] | [**a**] } { **+|-** } { **rwxt** } *File...* | *Directory...*

Changes permission modes in symbolic form.

**-R** Sets modes for each file in a directory.

**u** User (owner)

**g** Group

**o** Others

**a** All of the above

**+** Grants permission

**-** Removes permission

**r** Read

**w** Write

**x** Execute (for files), search (for directories)

**t** Save text (for files), link (for directories)

```
chmod go-w+x directory1
```

Denies to group and others write permission in `directory1`. Grants to group and others search permission in `directory1`.



### **chmod** *OctalNumber* *File...* | *Directory...*

Changes permission modes in numerical form.

<b>OctalNumber</b>	<b>Permission Numbers</b>
4000	Sets user ID on execution.
2000	Sets group ID on execution.
1000	Sets link permission to directories or the save text attribute for files.
0400	Permits read by owner.
0200	Permits write by owner.
0100	Permits execute or search by owner.
0040	Permits read by group.
0020	Permits write by group.
0010	Permits execute or search by group.
0004	Permits read by others.
0002	Permits write by others.
0001	Permits execute or search by others.

*OctalNumber* is the sum of the numbers representing the desired permissions.

```
chmod 0750 directory1
```

Grants read, write and search permissions in *directory1* to the owner. Grants read and search permissions in *directory1* to the group. Denies all permissions to others.

## Updating File Modification and Access Times

**touch** [ **-a -c -f -m** ] [ *Time* ] { *Directory...* | *File...* }

Updates access and modification times of a file.

- a** Changes only access time.
- c** Does not create files if they do not already exist.
- f** Tries to force touch despite restrictions on file.
- m** Changes only modification time.



*Time* is in *MMDDHHmm*  
(month-day-hour-minute) format.

```
touch -c 10312400 file1
```

Sets access and modification times of *file1*  
to midnight, October 31. Does not create *file1*  
if it does not already exist.

## Editing Files

**e** [ *.* ] [ [ *File* [ *Line* [ *Column* [ *SearchKey* ] ] ] ] ]

Starts the INed editor to create or update ASCII  
files.

- .** Opens File Manager.
- File* Starts INed editing *File* at the location  
specified.
- e** When used alone, brings up last file  
edited.

To find out which keys on your keyboard are  
command keys corresponding to INed editor  
functions, enter the **keymaps** command.

**ed** [ *File* ]

Starts the ed line editor to create or update ASCII  
files.

For more information, see “Using the ed Editor” on  
page 36.

**sed** [ *Pattern* ] [ **-n -e Script... -f SourceFile...** ]  
[ *File...* ]

Provides a stream editor.

*Pattern* Search expression.

- n** Suppresses information normally  
written to standard output.
- e Script** Uses *Script* as the edit script.



**-f *SourceFile***

Uses *SourceFile* as the source of the edit script.

```
sed "s/red/green/g" file1 > newfile
```

Replaces every occurrence of `red` with `green`, putting the edited version in `newfile`. The `s` performs a search for a pattern. The `g` performs a global replacement.

**Note:** The above example would change `Fred` to `Fgreen`.

```
sed -n "/food/p" file1
```

Displays every line in `file1` containing the word `food`. `p` prints lines containing the specified pattern.

**vi** [-I -R -r [ *File* ]][ *File*...]

Enters **vi**, a full-screen editor.

**-I** Enters in Lisp mode.

**-R** Enters vi to read files without changing them.

**-r** [ *File* ] Recovers last saved edition of *File* after a system crash.

For more information, see “Using the vi Editor” on page 32.

## Printing

### Starting a Print Job

**qprt** [-I *Number* -m *Text* -N *Number* -P *Queue* -= *OutputBin*] *File*...]

Starts a print job.

**-I** *Number* Sets page length to *Number* lines.



**-m** *Text* Sends message to console when job is ready to print. Printing begins when message is acknowledged.

**-N** *Number*  
Sets *Number* of copies to be printed.  
Default is one.

**-P***Queue*  
Specifies the print queue to be used.

**-=***OutputBin*  
Specifies the output bin destination for a print job. Valid output bins are printer-dependent.

```
qprt myfile
```

Queues the *myfile* file to print on the default print queue, using the default values.

```
qprt -Php2 myfile
```

Queues the *myfile* file to print on print queue *hp2*, using the default values.

On AIX Version 4.3 or later, you can use the Web-based System Manager Print 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.

## Displaying Print Status

**qchk** [ **-A** ] [ **-P***Queue* ] [ **-#***JobNumber* ] [ **-u***User* ]

Displays the status of a print queue.

**-A** Displays the status of all print queues.

**-P***Queue* Displays the status of the specified print queue.

**-#***JobNumber*  
Displays the status of the specified job number.



**-uUser** Displays the status of all print jobs sent by the specified user.

```
qchk -Php2
```

Displays the status for print queue `hp2`.

On AIX Version 4.3 or later, you can use the Web-based System Manager Print 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.

## Canceling a Print Job

```
qcan [-x JobNumber] [-PQueue]
```

Cancels a print job.

**-x JobNumber**

Specifies that only that particular *JobNumber* be canceled.

**-PQueue**

Specifies the print queue.

```
qcan -x 756 -Php2
```

Cancels job number `756` on print queue `hp2`.

On AIX Version 4.3 or later, you can use the Web-based System Manager Print 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.

## Holding or Releasing a Print Job

```
qhld [-r] { -#JobNumber [-PQueue] | -PQueue  
| -uUser [-PQueue] }
```

Holds and then releases a spooled print job.





- r** Releases the print job by number, queue, or user name
- #JobNumber** Specifies the print job number to be held.
- PQueue** Specifies the print queue to be held.
- uUser** Specifies the name of user whose print jobs are to be held.

qhld -#452

Holds print job number 452.

qhld -#452 -r

Releases print job number 452.

## Moving a Print Job

**qmov -mNewQueue**  
{ **-#JobNumber** [ **-PQueue** ] | **-PQueue** | **-uUser** [ **-PQueue** ] }

Moves a print job to another print queue.

- #JobNumber** Specifies the job number of the print job to be moved.

- mNewQueue** Specifies the name of the destination print queue.

- PQueue** Specifies the present print queue of the job to be moved.

- uUser** Specifies the name of user whose print jobs are to be moved.

qmov -mhp2 -#280

Moves job number 280 to print queue hp2.

qmov -mhp2 -Php4D

Moves all print jobs on print queue hp4D to print queue hp2.



## Working with Processes and Commands

### Scheduling Jobs to be Run Later

`at [-k -m ] Time [ Date ] [ Increment ] [ Job ]`

Schedules jobs to be run later.

**-k** Runs *Job* in Korn shell.

**-m** Mails message to user when *Job* is done.

*Time* is *hhmm* (hour minute) with optional **am** or **pm**. Default is 24-hour clock. Special times recognized are: **noon**, **midnight**, and (with *Date* or *Increment*) **now**.

*Date* is in *month name* and *day number* format or *day of the week* format. Special dates recognized are **today** and **tomorrow**.

*Increment* can be **+number** or **next** followed by **minute(s)**, **hour(s)**, **day(s)**, **week(s)**, **month(s)**, **year(s)**.

*Job* is the name of the command to be run rather than a parameter of the **at** command. The job must be in the current directory or the full path name to it must be given.

```
at 5:30 pm December 31 uuclean
```

Runs **uuclean** at 5:30 pm on December 31.

```
at now next week uuclean
```

Runs **uuclean** at this time next week.

### Listing or Removing Jobs

`at { -l | -r [Job...] }`

Lists or removes jobs from the schedule.

**-l** Lists scheduled *Jobs*.

**-r** [*Job...*] Removes *Job* from schedule. Default is all jobs.

## Displaying Environment Variables

### **env**

Displays the current environment variables.

## Stopping a Process

**kill** [ *-Signal* ] *ProcessID...*

Sends a signal to *ProcessID*. The default signal terminates the process.

```
kill 1095
```

Terminates process 1095.

```
kill -9 201
```

Kills (*signal 9*) the group of processes associated with *ProcessID 201*.

## Reading and Sending Mail

**mail** [ *-f* [ *PathName* ] ] | **mail** *address...*

Reads and sends mail. This command displays the mailbox prompt ( > ).

**-f** [ *PathName* ]

Displays a listing of the messages in your mailbox.

*address* Address of user to whom you wish to send mail.

## Changing Your Password

### **passwd**

Changes your password.



## Showing Status of Processes

**ps** [ **-e** ] [ **-f** | **-l** | **u** ] [ **-u** *Ulist* ]

Shows current status of processes. *Ulist* is a list of user names separated by commas.

- e** Shows all processes except kernel processes.
- f** Generates a full listing.
- l** Generates a long listing.
- u** Displays user-oriented output.
- u** *Ulist* Displays information about processes owned by users in *Ulist*.

```
ps -e -f
```

Displays all processes.

```
ps -f -l -u user1,user2,root
```

Lists processes owned by `user1`, `user2`, and `root`.

## Shutting Down the System

**shutdown** [ **-d -F -h -i -k -r** ] [ **+** *TimeMessage* ]

Ends system operation.

- d** Brings system from distributed mode to multiuser mode.
- F** Performs a fast shutdown, bypassing messages to other users.
- h** Halts system completely.
- i** Displays interactive messages to help user through shutdown.
- k** Avoids shutting down system.
- r** Reboots system.

**+** *TimeMessage*

Either *Number* (of minutes till shutdown) or *Hour:Minute* (time of shutdown). Default is one minute.



## Setting Workstation Parameters

**stty** [ **-a -g** ] [ **sane** ] [ *options* ]

Sets, resets, or reports workstation operating parameters.

- a** Writes the current state of all option settings to standard output.
- g** Writes option settings to standard output in a form usable by another **stty** command.
- sane** Resets parameters to default values.

## Accessing a Remote Host

**tn** [ **-d -n** *TraceFile* **-e** *TerminalType* ] [ *Host* [ *Port* ] ]

Logs into a remote host.

- d** Turns debug mode on.
- n** *TraceFile*  
Records trace information in *TraceFile*.
- e** *TerminalType*  
Sets terminal type. Options are: vt100, 3270, none.
- Host** Name or address of machine to log in to.
- Port** Specific port address to log in to.

`tn host1`

Logs into remote host `host1` and performs terminal negotiation.

## Displaying the Operating System Name

**uname** [ **-a** ] [ [ **-m -n -r -s -v** ] ]

Displays the name of the current operating system.



- a** Reports all information of the flags described below.
- m** Hardware ID number.
- n** Node name.
- r** Operating system release number.
- s** System name (on by default).
- v** Operating system version.

## Identifying Logged-In Users

**who** [-a ] | [ -d -p -r -u ] [ *File* ] |  
**who am i**

Identifies users currently logged in.

- a** Includes all flags.
- d** Reports expired processes.
- p** Reports previously spawned processes.
- r** Reports run level of process.
- u** Reports users and workstations active.

`who am i`

Displays your user name.

## Working with Shells

- bsh** Starts the Bourne shell.
- csh** Starts the C shell.
- ksh** Starts the default login shell, the Korn shell.
- sh** Starts the default shell, the Korn shell.



## Using Web-based System Manager on AIX Version 4.3 or Later

The Web-based System Manager is a graphical user interface (GUI) application available on AIX Version 4.3 or later for managing your system. You can manage systems from a locally attached display or remotely from another AIX system or personal computer equipped with a web browser. You can perform such tasks as viewing users and groups, installed software, and printers and devices; managing logical volumes; mounting and unmounting file system; configuring the network; as well as numerous other tasks.

Provides point-and-click control of objects, which provides an alternative to learning and using AIX commands.

For additional information, see page 50, refer to *AIX Version 4.3 AIX Management Guide: Operating System and Devices*, or see the online help.

## Using SMIT

**smit** [ **-D -f -h** ] [ *FastPath* ]

Starts SMIT (System Management Interface Tool) to perform system management tasks.

- D** Turns on debug mode.
- f** Allows redirection of standard input and standard output from SMIT.
- h** Displays command usage message.

*FastPath* Allows you to enter SMIT below the main menu. *FastPath* parameters are SMIT commands.



## Getting Information and Help

### date

Displays current date and time.

### help

Provides a one-page display of information for new users.

## Accessing Online Information

### HTML-Based AIX Documentation

The online information for AIX Version 4.3 or later is available on CD-ROM. After the CD is mounted or its contents installed, you can view the online information through an HTML version 3.2-compliant web browser. If you do not have a web server installed, the search function will not work since it requires a web server.

The *AIX Version 4.3 Base Documentation* CD contains most of the base operating system books and can be viewed with a web browser. To access the information, do one of the following:

- If you are using the CDE Desktop, open the Help subpanel and click on the AIX Base Documentation icon.
- Enter the location of the documents into your web browser's URL field. If you do not have a web server installed, request the following URL:

```
file:///usr/share/man/info/en_US/  
a_doc_lib/aixgen/topnav.htm
```

If you have a web server installed, request the following URL:

```
http://Your_System_Name/doc_link/  
/en_US/a_doc_lib/aixgen/topnav/  
topnav.htm
```

For more information, see *AIX Version 4.3 Release Notes* or *AIX Version 4.3 Quick Beginnings*.





## InfoExplorer

The online information for AIX Versions 4.1 and 4.2 only is available through InfoExplorer. To start InfoExplorer, type:

```
info [ -h Label ] [-l LibraryName | -n { bl cl ed ti } ]  
[ -s String ]
```

**-h *Label*** Specifies a help string to search on.

**-l *LibraryName***  
Specifies a database library.

**-n** Specifies the default navigation article. Options are:

<b>bl</b>	Book Contents List
<b>cl</b>	Commands List
<b>ed</b>	Education Menu
<b>ti</b>	Task Index

**-s *String*** Specifies a string for which you wish to search.

```
info -h grep
```

Starts InfoExplorer and displays the **grep** article.

For additional information, see *AIX Version 4 Getting Started*.

## Backing Up and Restoring

### Backing Up Files

```
backup [ -Level -u ] [ FileSystem ]
```

Backs up files by i-node.

**-Level** Specifies level of backup (0-9).  
Default is 9.

**-u** Updates information for incremental backup.

```
backup -0 -u
```

Backs up files changed since last level 0 backup, and gives progress reports.



On AIX Version 4.3 or later, you can use the Web-based System Manager Backups application (**wsm backup** fast path) to run this command.

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

#### **backup -i [-v ] [-f dev]**

Backs up files by name.

**-v** Lists files as they are backed up.

**-f dev** File or device to back up data to.

```
find $HOME -print | backup -i -v  
-f /tmp/backup1
```

Lists selected files from the **\$HOME** directory and backs them up to the file `/tmp/backup1`.

On AIX Version 4.3 or later, you can use the Web-based System Manager Backups application (**wsm backup** fast path) to run this command.

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

## Compressing Files

#### **compress [-f] [-c] [-v]**

Compresses data in a file. The compressed file retains the same ownership, modes, and modification time of the original file. The specified file is replaced by a compressed file with a **.Z** appended to its name. Typically, source code or English text is reduced by 50 to 60%.

**-f** (or **-F**) Forces compression or expansion.

**-c** Causes compress to write to standard output. No files are changed.



**-v** Writes the percentage of compression.

```
compress -v myfile
```

Compresses `myfile` and writes the percentage of compression to standard error.

### **uncompress [-f] [-c]**

Restores original files that were compressed by the **compress** command.

**-f** (or **-F**) Forces compression or expansion.

**-c** Causes uncompress to write to standard output. No files are changed.

```
uncompress myfile
```

Uncompresses `myfile`.

## **Restoring Previously Backed-Up Files**

**restore [-r | -T] [-f dev] [-v -x] [-i | -d] [File]**

Restores previously backed-up file systems or files created by the **backup** command.

**-r** Restores an entire file system. Do not use with *File* parameter.

**-T** Displays table of contents for backed-up files. Do not use with *File* parameter.

**-f dev** File or device to restore previously backed-up data from.

**-v** Reports progress of restoration.

**-x** Restores individually named files.

**-i** Starts interactive mode, to restore selected files from the backup.

**-d** If *File* is a directory, all files in it are restored.



```
restore -T -f /tmp/backup1
```

Lists names of files previously backed up to file /tmp/backup1.

```
restore -x -v
```

Restores files to main file system.

```
restore -x -d -v Directory1
```

Restores `Directory1` and all files in it.

## Writing Files To or From Storage

```
tar { -c | -r | -t | -u | -x } [ -v -w ] -f Archive ] [ File... | Directory... ]
```

Manipulates archives by writing files to, or retrieving files from, an archive storage medium.

- c** Creates a new archive.
- r** Appends files to an archive.
- t** Lists names of files in an archive.
- u** Adds files to an archive if they are not already there or have been modified.
- x** Extracts files from an archive.
- v** Lists name of each file as it is processed.
- w** Waits for user confirmation.
- f** *Archive* *Archive* to be read from or written to.

```
tar -c file1 file2
```

Writes `file1` and `file2` to the default backup device.

```
tar -v -t -f out.tar
```

Displays names of files in archive `out.tar`.

```
mount [ all ] | [ -n Node -t Type -v VfsName ]
```

Makes a file system available for use.

**all** Mounts all default file systems.



- n Node** Specifies remote node that holds directory to be mounted.
- t Type** Mounts all unmounted stanzas in the **/etc/filesystems** file containing the **type = Type** attribute.
- v VfsName**  
Specifies that the file system is defined by the *VfsName* parameter in the **/etc/vfs** file.

```
mount -n nodeA /u/a.remote  
/u/a.local
```

Mounts the remote file system `/u/a.remote` from remote host `nodeA` to the local file system `/u/a.local`.

On AIX Version 4.3 or later, 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 mount** fast path to run this command.

**umount** [ **-f** ] { **-a** | **all** | **-n Node** | **-t Type** | *File* | *Directory* | *FileSystem* }

Unmounts a previously mounted file, directory, or file system.

- f** Forces unmount of a remote file system.
- a** Unmounts all mounted file systems.
- all** Unmounts all mounted file systems, except those with **mount = automatic** attribute.
- n Node** Unmounts all remote mounts made from *Node*.
- t Type** Unmounts all files and directories of type *Type*.

On AIX Version 4.3 or later, 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 mount** fast path to run this command.

## Analyzing Space Utilization

**df** [ *FileSystem...* | *File...* ]

Displays information about space on all mounted file systems.

```
df .
```

Displays available space on current file system.

**du** [ **-r -s** ] [ **-a** [ *File...* ] ] [ *Directory...* ]

Summarizes disk usage.

**-r** Indicates inaccessible files and directories.

**-s** Displays sum total disk usage for specified files or directories.

**-a** [*File...*] Displays disk use for each file.

```
du -a Dir1
```

Displays disk usage for each file in the directory specified by *Dir1*.

## Solving Problems

**acctcom** [ **-u***User* ] [ **-n***Pattern* ] [ **-s** *Time* ] [ *File...* ]

Displays selected process accounting record summaries. If you specify a *File*, the **acctcom** command reads each file chronologically by process completion time.



- u***User* Shows only processes belonging to the specified *User*.
- n***Pattern* Shows only commands matching the value of *Pattern*, where *Pattern* is a regular expression.
- s***Time* Shows only those processes that existed on or after the specified *Time*.

```
acctcom -u smith -s 17:00
```

Displays information about processes that belong to user `smith` and that run after 5 p.m., using the default accounting data file.

#### **iostat** [-d] [-t] [ *Interval* [ *Count* ]

Reports central processing unit (CPU) statistics and input/output statistics for tty devices, disks, and CD-ROMs. This command generates reports that can be used to change system configuration to better balance the input/output load between physical disks.

- d** Displays only the disk utilization report. Exclusive of the **-t** option.
- t** Displays only the tty and cpu usage reports. Exclusive of the **-d** option.
- Interval* Specifies amount of time in seconds between each report.
- Count* Number of reports generated. Can only be specified with the *Interval* parameter.

```
iostat -d disk1 2
```

Displays a continuous disk report at two-second intervals for the disk with the logical name `disk1`.

#### **sar** [-c] [-u] [ *Interval* [ *Number* ]

Collects, reports, or saves system activity information.

- c** Reports system calls.



- u** Reports processor or system-wide statistics.
- Interval* Specifies amount of time in seconds between each report.
- Number* Number of reports generated.

```
sar -r 2 10
```

Reports current activity for each two seconds, for a total of ten reports.

On AIX Version 4.3 or later, you can use the Web-based System application (**wsm system** fast path) to run this command.

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

#### **vmstat** [ *Interval* [ *Count* ] ]

Reports statistics about kernel threads, virtual memory, disks, traps, and CPU activity. Reports generated by **vmstat** can be used to balance system load activity.

- Interval* Specifies amount of time in seconds between each report.
- Count* Number of reports generated. Can only be specified with the *Interval* parameter.

```
vmstat 2 5
```

Displays statistics for each two seconds, for a total of ten reports.





---

## Using Command Line Key Functions

Function	Key Sequence
Erase entire line.	Ctrl-U
Erase preceding character.	Backspace
Erase word preceding cursor.	Ctrl-W
Interrupt.	Ctrl-C
Log out and terminate file input.	Ctrl-D
Quit program and create a file named <b>core</b> that is used for debugging.	Ctrl-\
Resume output of program stopped by Ctrl-S.	Ctrl-Q
Start running a command after you have finished entering the text.	Enter
Stop output from running off the bottom of the screen.	Ctrl-S



---

## Using the vi Editor

With the vi editor, you can edit files using a full-screen display.

### Entering vi

`vi [-l -R -r [ File ]][ File... ]`

- `-l` Enters in Lisp mode.
- `-R` Enters **vi** to read files without changing them.
- `-r [ File ]` Recovers last saved edition of *File* after a system crash.

The vi editor has two modes: command and insert. The editor opens in command mode.

### Command Mode

- `Esc` Ends Input mode or incomplete subcommand.
- `Ctrl-?` Interrupts a subcommand.
- `Ctrl-G` Finds file information, including number of line on which cursor is located.
- `!command`  
Runs an operating command in **vi**. Pressing Enter returns you to **vi**.
- n*  
Many commands may have an *n* prefixed to them to perform the action *n* times. The default is one.
- `:set attribute`  
Allows you to set the following attributes:
  - `nu` Line numbering
  - `list` Show hidden characters
  - `showmode` Mode display
  - `ts=` Tabs



Prefacing an attribute with **no** will suppress it.

### Cursor Movement

<b>O</b>	Returns to beginning of line.
<b>h</b>	Space left.
<b>l</b>	Space right.
<b>k</b>	Line up.
<b>j</b>	Line down.
<b>w</b>	Word forward.
<b>b</b>	Word backward.
<b>)</b>	Sentence forward.
<b>(</b>	Sentence backward.
<b>H</b>	Top of screen.
<b>M</b>	Middle of screen.
<b>L</b>	Bottom of screen.
<b>G</b>	Last line.
<b>nG</b>	Line <i>n</i> .

### Screen Scrolling and Paging

Ctrl-E	Scrolls down by line.
Ctrl-D	Scrolls down by half screen.
Ctrl-F	Scrolls down by full screen.
Ctrl-Y	Scrolls up by line.
Ctrl-U	Scrolls up by half screen.
Ctrl-B	Scrolls up by full screen.
<b>z+</b>	Pages down.
<b>z^</b>	Pages up.

### Changing Text

<b>C</b>	Changes rest of line.
<b>cc</b>	Changes a line.
<b>cw</b>	Changes a word.
<b>J</b>	Joins lines.



<b>rx</b>	Replaces character with <i>x</i> .
<b>Rstring</b>	Replaces text beginning at cursor with <i>string</i> .
<b>s</b>	Substitutes character.
<b>S</b>	Substitutes line.
<b>D</b>	Deletes rest of line.
<b>dd</b>	Deletes current line.
<b>n dd</b>	Deletes <i>n</i> lines.
<b>dw</b>	Deletes a word.
<b>x</b>	Deletes character at cursor.
<b>X</b>	Deletes character before cursor.
<b>u</b>	Undoes last change.
<b>U</b>	Restores current line if cursor has not left it since last change.
<b>.</b>	Repeats last change.
<b>~</b>	Switches case of letter at cursor.
<b>:g;Text;s;;Text2:g</b>	Replaces all occurrences of <i>Text</i> with <i>Text2</i> .

### Moving and Copying

<b>y(object)</b>	Yanks (copies) <i>object</i> into undo buffer.
<b>yw</b>	Yanks (copies) word into undo buffer.
<b>yy</b>	Yanks (copies) current line into undo buffer (same as <b>Y</b> ).
<b>p</b>	Inserts text from undo buffer after cursor or after current line.
<b>P</b>	Inserts text from undo buffer before cursor or before current line.



## Saving and Quitting

- :w** [*File*] Saves edited file to *File* if specified.
- ZZ** Saves file and quits.
- :wq** Saves file and quits.
- :q** Quits. If file has been changed, displays warning before quitting.
- :q!** Quits without saving changes.

## Insert Mode

To enter Insert mode, type one of the following characters: **a**, **A**, **i**, **I**, **o**, **O**.

- a***Text* Inserts *Text* after cursor.
- A***Text* Inserts *Text* at end of line.
- i***Text* Inserts *Text* before cursor.
- I***Text* Inserts *Text* at beginning of line.
- o** Adds empty line below current line and puts you in input mode.
- O** Adds empty line above current line and puts you in input mode.
- Ctrl-H or Backspace Erases last character.
- Ctrl-W Erases last word.
- Esc Ends insert mode and returns to command mode.



---

## Using the ed Editor

The ed editor edits text a line at a time.

### Entering ed

- ed** [*File*] Enters **ed**.
- h** Displays help message for current ?, diagnostic or error message.
- H** Turns help messages on or off for the session.
- P** Turns \* (asterisk) cursor on or off.
- !** *Command* Runs an operating system command.
- w** Writes to file.

### Addressing Lines

- Number* Addresses line *Number*.
- .** Addresses current line.
- 0** Addresses before first line of file.
- \$** Addresses last line.
- /Pattern/** Addresses line containing *Pattern*.
- =** Displays address.

### Searching

- /Pattern** Searches forward for *Pattern*.
- ?Pattern** Searches backward for *Pattern*.
- g/Pattern/** Selects lines containing *Pattern*.
- v/Pattern/** Selects lines not containing *Pattern*.

### Displaying Text

- p** Prints lines addressed.
- l** Prints lines, representing normally non-printing characters visibly.



**n** Prints lines including line number.

## Command Mode

**s/***OldPattern*/*NewPattern*/ [ **g** ]  
Substitutes *NewPattern* for  
*OldPattern*. **g** causes global  
substitution.

**s/***Pattern*// Deletes *Pattern* from current line.

**d** Deletes current line.

**m** [ *Address* ]  
Moves current line after given  
address.

**t** [ *Address* ]  
Copies current line after given  
address.

**u** Undoes last change to text.

## Insert Mode

**a** Appends text after addressed line.

**i** Inserts text before addressed line.

**c** Deletes addressed lines and replaces  
them with new input.

**.** Returns to command mode.

## Leaving ed

**q** Quits and prompts to save changes if  
not already written to file.

**Q** Quits and discards changes.



---

## Working with the Korn Shell

The Korn shell (also known as the POSIX shell) is an interactive command interpreter and a command programming language. The Korn shell is the default shell for the AIX operating system. The shell carries out commands either interactively from a workstation keyboard or from a file.

### Vocabulary

#### built-in command

A command that the shell executes without searching for it and creating a separate process.

#### command

A sequence of characters in the syntax of the shell language. The shell reads each command and carries out the desired action either directly or by starting separate utilities.

#### comment

Any word that begins with a # (pound sign). The word and all characters that follow it, until the next new-line character, are ignored.

#### identifier

A sequence of letters, digits, or underscores from the portable character set, starting with a letter or underscore.

#### job control

A facility for selectively stopping the execution of processes and then continuing their execution at a later point.

#### list

A sequence of one or more pipelines, separated by one of these symbols ; (semicolon), & (ampersand), && (double ampersand), or || (double bar). It can be ended by ; or & or |&.





**; & ( ) | \$ \ < > \ ' ' newline space tab**

These are *metacharacters*. Each metacharacter has a special meaning to the shell and, unless quoted, causes termination of a word. Quoting mechanisms are the \ (backslash), single quotes, and double quotes.

**parameter assignment list**

Includes one or more words of the form *Identifier=Value*. Spaces surrounding the equal sign must be balanced.

**pipeline**

A sequence of one or more commands separated by a | (pipe). Each command in a pipeline, except possibly the last command, runs as a separate process. The standard output of each command connected by a pipe becomes the standard input of the next command in the sequence.

**shell variable**

A name or parameter for which a value can be assigned. Assign a variable by typing the variable name, an = (equal sign), and then the value (*Name = Value*). The variable name can be substituted for the assigned value by preceding the variable name with a \$ (dollar sign). For example, *\$Name*.

**simple command**

A sequence of optional parameter assignment lists and redirections, in any sequence. They are optionally followed by commands, words, and redirections. They are terminated by ;, |, &, ||, &&, |&, or a new-line character.

**subshell**

A shell that is running as a child of the login shell or the current shell.

**wildcard character**

Also known as a pattern-matching character. The shell associates them with



assigned values. The basic wildcards are `?`, `*`, `[set]`, and `!set`. Particularly useful when performing file name substitution.

**word** A sequence of characters that does not contain any blanks. Words are separated by one or more non-quoted metacharacters.

## Starting and Stopping the Shell

**ksh** Starts the Korn shell.

**exit** Terminates the shell.

**Ctrl-D** Terminates the shell.

**Ctrl-C** Sends interrupt signal to the foreground process group.

## Creating Shell Scripts

A shell script is a file containing shell commands. Create the script using any standard editor. Run the script by entering the shell command and the script file name. For example, to run a nonexecutable file named `script1` under the control of the Korn shell, type `ksh script1`. You can also make the file executable with **chmod +x** to run the script by name. See **chmod** command on page 9.

## Quoting

When you want the Korn shell to read a character as a regular character, you must quote it. You can use one of these quoting characters:

- `\` Quotes a single character.
- `' '` Single quotes. Quotes enclosed string except for `'` itself.
- `""` Double quotes. Quotes enclosed string except for `&`, ``` (back quote), and `\`.



## Reserved Words

The following reserved words have special meaning to the shell:

<b>!</b>	<b>case</b>	<b>do</b>
<b>done</b>	<b>elif</b>	<b>else</b>
<b>esac</b>	<b>fi</b>	<b>for</b>
<b>function</b>	<b>if</b>	<b>in</b>
<b>select</b>	<b>then</b>	<b>time</b>
<b>until</b>	<b>while</b>	<b>{</b>
<b>}</b>	<b>[[</b>	<b>]]</b>

The reserved words are recognized only when they appear without quotation marks and when the word is used as the:

- First word of a command
- First word following one of the reserved words other than **case**, **for**, or **in**
- Third word in a **case** or **for** command (only **in** is valid in this case).

## Command Aliasing

The Korn shell enables you to create aliases to customize commands. The **alias** command defines a word of the form `Name=String` as an alias. When you use an alias as the first word of a command line, **ksh** checks to see if it is already processing an alias with the same name. If it is, **ksh** does not replace the alias name. If an alias with the same name is not already being processed, **ksh** replaces the alias name by the value of the alias.

## Tilde Substitution

After alias substitution is done, the shell checks each word to see if it begins with an unquoted `~` (tilde). If it does, the shell checks the word, up to the first `/` (slash), to see if it matches a user name in the `/etc/passwd` file. If it does, the `~` and the matched login name are replaced by the login directory of the matched user.



## Parameter Substitution

A *parameter* is categorized as an:

- Identifier of any of the characters `*`, `@`, `#`, `?`, `-`, `$`, and `!`. These are called *special parameters*.
- Argument denoted by a number. This is a *positional parameter*.
- Parameter denoted by an identifier and has a *value* and zero or more *attributes*. This is a *named parameter*. Named parameters can be assigned values and attributes by using the **typeset** special built-in command.

## Command Substitution

With command substitution, the shell executes a specified command in a subshell environment and replaces that command with its output.

## Pattern Matching

In the Korn shell, you can use wildcard characters to match any string of characters:

- `*` Matches any string, including the null string.
- `?` Matches any single character.
- `[ . . . ]` Matches any one of the enclosed characters.
- `[.-]` Matches any character between the enclosed pair, including the pair.
- `[! . . . ]` Matches any single character except one of the enclosed.
- `?(PatternList)` Matches any one of the given patterns.
- `*(PatternList)` Matches zero or more occurrences of the given patterns.
- `+(PatternList)` Matches one or more occurrences of the given patterns.



@(*PatternList*)

Matches exactly one of the given patterns.

!(*PatternList*)

Matches anything except one of the given patterns.

## Redirecting Input and Output

Before a command is executed, its input and output can be redirected using special notation which can appear anywhere in a simple command. Command and parameter substitution occurs before the file *Word* is used.

**<Word** Uses file *Word* as standard input to a command.

**>|Word** Writes standard output to file named *Word*. Use **>|** to override the **noclobber** option.

**>>Word** Writes to a file named *Word* as standard output. If that file exists, then output is appended to it.

**<>Word** Opens file *Word* for reading and writing as standard input.

**<<[-]Word** Enables redirection of lines contained in a shell input file, known as a here-document, to the standard input of a command.

**<&Word** Duplicates one input file descriptor from another, or to close one.

**>&Word** Duplicates one output file descriptor from another, or to close one.

## Command Grouping

You can create a subshell by running commands enclosed in ( ).

You can run commands as a unit by enclosing them in { }.



## Using Built-In Commands

Certain commands are built in to the Korn shell and executed in the shell process. There are two types of built-in commands: *special built-in commands* and *regular built-in commands*.

### Special Built-In Commands

**:** [*Argument ...*]

Expands only arguments.

**.** *File* [*Argument ...*]

Reads the complete specified file and then executes the commands in the current shell environment.

**break** [*n*]

Exits from enclosing loop. If *n* is specified, then breaks *n* levels.

**continue** [*n*]

Resumes next iteration of enclosing **for**, **while**, **until**, or **select** loop. If *n* is specified, then resumes at the *n*th enclosing loop.

**eval** [*Argument...*]

Reads the specified arguments as input to the shell and executes the resulting command or commands.

**exec** [*Argument...*]

Executes command specified by the argument in place of this shell, without creating a new process.

**exit** [*n*]

Exits the shell with the exit status specified by the *n* parameter. If *n* is omitted, the exit status is that of last command run.

**export** **-p** [*Name[=Value]*] ...

Marks specified names for automatic export to the environment of subsequently executed commands.

**newgrp** [*Group*]

Equivalent to the **exec /usr/bin/newgrp** [*Group...*] command.



**readonly** **-p** [*Name*[=*Value*]] ...

Marks the names specified by *Name* parameter as read-only. These names cannot be changed by subsequent assignment

**return** [*n*] Causes a shell function to return to the invoking script. Return status is specified by the *n* variable.

**set** Sets environment for the shell.

**shift** [*n*] Renames the positional parameters, beginning with  $\$_{n+1}$  ... through  $\$1$  ... The default value of the *n* parameter is 1.

**times** Prints the accumulated user and system times for the shell and for processes run from the shell.

**trap** [*Command*] [*Signal*] ...

Runs the specified command when the shell receives the specified signal or signals. The *Signal* parameter can be given as a number or as the name of the signal.

**typeset** [**+HLRZfi**lrtux[*n*]] [**Name**[=*Value*]] ...

Sets attributes and values for shell parameters.

**unset** [**-fv**] *Name* ...

Erases the values and attributes of the parameters given by the list of names. If the **-f** flag is set, then the names refer to function names.

## Regular Built-In Commands

**alias** [**-t**] [**-x**][*AliasName* [= *String*]]

Creates or redefines alias definitions or writes existing alias definitions to standard output.

**bg** [*JobID*...]

Puts each specified job in the background.

**cd** [*Argument*]

**cd** *Old New*

This command can be in either of two forms. In the first form, it changes the



current directory to the one specified by the *Argument* parameter. The second form substitutes the string specified by the *New* variable for the string specified by the *Old* variable in the current directory.

**command** [-p] *CommandName* [*Argument...*]

**command** [-v / -V] *CommandName*

Causes the shell to treat the specified *CommandName* as a simple command, suppressing shell function lookup.

**echo** [*String ...*]

Writes character strings to standard output.

**fc** [-r] [-e *Editor*] [*First* [*Last*]] |

**fc** -l [-n] [-r] [*First* [*Last*]] |

**fc** -s [*Old=New*] [*First*]

Displays the contents of your command history file or invokes an editor to modify and re-executes commands previously entered in the shell.

**fg** [*JobID...*]

Brings each job specified into foreground.

**getopts** *OptionStringName* [*Argument...*]

Checks the *Argument* parameter for legal options. If you omit the *Argument* parameter, the positional parameters are used.

**jobs** [-l | -n | -p] [*JobID...*]

Lists status of jobs started in the current shell environment. If no *JobID* is specified, status of all jobs is displayed.

**kill** Sends a signal to a running process. The default action normally stops processes.

**let** *Expression*

Evaluates specified arithmetic expressions.

**print** [-Rnprsu[n]] [*Argument ...*]

Prints shell output. With no flags, prints the arguments on standard output.

**pwd** Equivalent to **print -r - \$PWD**.





**read** [-**prsu**[*n*]] [*Name?Prompt*] [*Name...*]  
Takes shell input, one line is read and broken up into fields using the characters in the variable as separators.

**setgroups**  
Executes the **/usr/bin/setgroups** command, which runs as a separate shell. There is one difference: the **setgroups** built-in command invokes a subshell, but the **setgroups** command replaces the currently executing shell.

**test** Same as [[*Expression*]].

**ulimit** [-**HSacdfmst**] [*Limit*]  
Sets or displays user-process resource limits as defined in the **/etc/security/limits** file.

**umask** [-**S**] [*Mask*]  
Determines file permissions.

**unalias -a**  
**unalias** *AliasName ...*  
Removes the definition for each alias name specified, or removes all alias definitions if the **-a** flag is used.

**wait** [*ProcessID...*]  
Waits for the specified job and terminates.

**whence** [-**pv**] *Name...*  
Indicates, for each name specified, how it would be interpreted if used as a command name.

## Command Line Editing

The Korn shell includes a vi-style editor, which is entered by pressing the Esc key. The default mode is input. Pressing the Esc key switches between input and command modes. Most cursor movement, search, and text editing commands take an optional parameter *n*, which is the number of times the action is to be repeated. This optional parameter normally precedes the command.



### Cursor Movement Commands

<i>nh</i>	Space left.
<i>nl</i>	Space right.
<i>nw</i>	Word forward.
<i>nb</i>	Word backward.
<i>nf c</i>	Finds the next character <i>c</i> .
<i>nF c</i>	Finds the previous character <i>c</i> .
<i>n;</i>	Repeats the last <i>f</i> or <i>F</i> command.
<i>n,</i>	Undoes the last <i>f</i> or <i>F</i> command.
<i>0</i>	Moves to start of line.
<i>^</i>	Moves to first nonblank character of line.
<i>\$</i>	Moves to end of line.

### Accessing Command History Commands

<i>nk</i>	Fetches the previous command.
<i>nj</i>	Fetches the next command.
<i>nG</i>	Fetches the <i>n</i> th command.
<i>/String</i>	Searches backward in history for a command containing <i>String</i> .
<i>?String</i>	Searches forward in history for a command containing <i>String</i> .

### Text Editing Commands

#### Insert Mode

<i>aText</i>	Inserts <i>Text</i> after cursor.
<i>AText</i>	Inserts <i>Text</i> at end of line.
<i>iText</i>	Inserts <i>Text</i> before cursor.
<i>IText</i>	Inserts <i>Text</i> at the beginning of line.



## Command Mode

<i>cn</i>	Changes <i>n</i> characters.
<i>cc</i>	Changes line.
<i>C</i>	Changes rest of line.
<i>rx</i>	Replaces character with <i>x</i> .
<i>RString</i>	Replaces text beginning at cursor with <i>String</i> .
<i>dn</i>	Deletes <i>n</i> characters.
<i>dw</i>	Deletes word.
<i>dd</i>	Deletes current line.
<i>D</i>	Deletes rest of line.
<i>nx</i>	Deletes character at cursor.
<i>nX</i>	Deletes character before cursor.
<i>u</i>	Undoes last change.
<i>U</i>	Restores current line if cursor has not left it since last change.
<i>n.</i>	Repeats last change.
<i>n~</i>	Switches case of letter.
<i>y(object)</i>	Yanks (copies) <i>object</i> into undo buffer.
<i>yw</i>	Yanks (copies) word into undo buffer.
<i>yy</i>	Yanks (copies) current line into undo buffer.
<i>np</i>	Inserts text in undo buffer after cursor.
<i>nP</i>	Inserts text in undo buffer before cursor.



---

## System Management Tasks Using Web-based System Manager on AIX Version 4.3 or Later

With Web-based System Manager on AIX Version 4.3 or later, you can manage AIX systems from anywhere in the internet/intranet. Web-based System Manager is a complete system management environment that takes advantage of the latest in web technology.

Applications provide the Web-based System Manager with the power to manage the AIX system. The base system provides the user with 12 applications that can be run in local, remote, or applet (browser-based) mode.

You can start Web-based System Manager in the following ways:

- On an AIX system with a graphics terminal:
  - From the Common Desktop Environment (CDE) Application Manager
  - From a command line terminal window in the AIXwindows environment
  - From a command line terminal in CDE
- On a personal computer with an HTML version 3.2-compliant web browser

### From the CDE Application Manager

Select the Application Manager icon on the front panel. In the Application Manager window, select **System\_Admin**. In the Application Manager – System\_Admin window, select the Web-based System Manager icon. Clicking on the icon opens the launch pad.



### From the Command Line

If you are running the AIXwindows environment, open an xterm window. If you are running the CDE environment, open a dtterm window. On AIX Version 4.3 or later, you can start a Web-based System Manager application by typing its fast path command at the command line.

### From a Web Browser

On AIX Version 4.3 or later, you can open Web-based System Manager from a web browser by typing the following in your browser's URL or location field:

```
http://Your_Server_Name/wsm.html
```

This opens the Web-based System Manager launch container. Double click on the icon of the application that you want to use.

### Applications, Contents, and Fast Path Commands

On AIX Version 4.3 or later, the Web-based System Manager applications and their fast path commands are:

- **Backups**

Backs up your system

Fast path: **wsm backup**

- **Devices**

Physical and logical devices

Fast path: **wsm devices**

- **File Systems**

File systems installed on the system

Fast path: **wsm fs**

- **Logical Volumes**

Volume groups, logical volumes, and physical volumes

Fast path: **wsm lvm**



- **Network**

Network communications, services, and protocols

Fast path: **wsm network**

- **NIM**

Application for distributing software installation operations across a network

Fast path: **wsm nim**

- **Printer Queues**

Printers and print queues

Fast path: **wsm printers**

- **Processes**

Processes active on the system

Fast path: **wsm processes**

- **Software**

Software installed on the system

Fast path: **wsm software**

- **Subsystems**

Subsystem groups, subsystems, and subservers (daemons) on the system

Fast path: **wsm subsystems**

- **System**

System environment setup such as system console, user interface, and system date and time

Fast path: **wsm system**

- **Users**

Users and groups

Fast path: **wsm users**

For additional information, see *AIX Version 4.3 Quick Beginnings* or the online information.



---

## Using InfoExplorer Key Functions on AIX Versions 4.1 and 4.2

The tables below contain key functions for the ASCII version of InfoExplorer for AIX Version 4.1 and 4.2 only. For the windows interface, use the mouse to select window and menu options.

### Menu Bar

Function	Key Sequence
Move into or out of menu bar.	Ctrl-O
Close menu bar pull-down and move into text area.	Esc-Esc or Ctrl-O
Go to next menu bar option.	Tab or →
Go to previous menu bar option.	←
Perform selected operation.	Enter or Return
Go to previous option in a pull-down menu.	↑
Go to next option in a pull-down menu.	↓

### Pop-Up Panels

Function	Key Sequence
Move into or out of menu bar.	Ctrl-O
Go to next menu bar option.	→
Go to previous menu bar option.	←
Go to next field in pop-up panel.	Tab
Perform selected operation.	Enter or Return
Go to previous item in a list.	↑



Function	Key Sequence
Go to next item in a list.	↓
Cycle through options in a ring buffer.	Spacebar

## Text Area

Function	Key Sequence
Move into or out of menu bar.	Ctrl-O
Move between navigation and reading screens.	Ctrl-W
Select current link.	Enter or Return
Refresh screen.	Ctrl-L
Go to next link.	Ctrl-F or Tab
Go to previous link.	Ctrl-B
Scroll to previous screen.	Page Up or Ctrl-P
Scroll to next screen.	Page Down or Ctrl-N
Scroll text 20 characters to left.	<
Scroll text 20 characters to right.	>
Move cursor up one line in an article.	↑
Move cursor down one line in an article.	↓
Move cursor one character to left.	←
Move cursor one character to right.	→





## InfoExplorer Note Editor

Function	Key Sequence
Enter Insert mode.	a, A, i, I, o, O
Ends Insert mode.	Esc
Saves note and quits.	ZZ
Quits without saving changes.	:q!
Move a word forward.	h
Move a word backward.	l
Move a sentence forward.	)
Move a sentence backward.	(
Move to top of screen.	H
Move to bottom of screen.	L
Move to last line.	G
Scroll down a half screen.	Ctrl-D
Scroll down a full screen.	Ctrl-F
Scroll up a half screen.	Ctrl-U
Scroll up a full screen.	Ctrl-B
Move to last line.	G
Changes rest of line.	C
Changes line.	cc
Changes word.	cw
Deletes rest of line.	D
Deletes line.	dd
Deletes word.	dw
Deletes character at cursor.	x
Undoes last change.	u
Repeats last change.	.
Switches case of letter.	~

L



---

# Index

## A

acctcom command, 28  
at command, 16

## B

backup command, 23  
Bourne shell, 20  
bsh command, 20  
built-in commands, Korn shell, 44

## C

C shell, 20  
cat command, 1  
cd command, 2  
chgrp command, 8  
chmod command, 9  
chown command, 9  
command line key functions, 31  
compress command, 24  
cp command, 2  
csh command, 20

## D

date, displaying current, 22  
date command, 22  
del command, 2  
df command, 28  
diff command, 3  
directories  
    changing current, 2  
    changing ownership, 8, 9  
    changing permissions, 9  
    contents, listing, 6  
    copying, 2  
    creating, 7  
    deleting, 3, 8  
    displaying current path, 7  
    moving, 7  
    removing, 8



- renaming, 7
- sorting contents, 4
- unlinking, 3

disk usage, summarizing, 28

documentation, viii, 22

du command, 28

## **E**

e command, 11

ed command, 11

ed editor, 36

editors

- ed, 11, 36
- INed, 11
- sed, 11
- vi, 12, 32

env command, 17

environment variables, displaying, 17

## **F**

files

- backing up, 23
- changing ownership, 8, 9
- changing permissions, 9
- comparing, 3
- compressing, 24
- concatenating, 1
- copying, 2
- deleting, 2, 3
- displaying, 1
- displaying by page, 1
- displaying lines of, 5
- editing, 11
- finding, 4
- linking, 6
- moving, 7
- renaming, 7
- restoring backed-up, 25
- searching for patterns in, 5
- transferring, 4
- uncompressing, 25
- unlinking, 3
- unmounting, 27
- updating access times, 10
- writing to or from storage, 26



- writing to standard output, 8

find command, 4

ftp command, 4

## **G**

grep command, 5

## **H**

head command, 5

help, getting, 22

help command, 22

highlighting, vii

HTML, viii, 22, 50

## **I**

INed editor, 11

info command, 23

InfoExplorer, starting, 23

InfoExplorer key functions, 53

information, getting, 22

iostat command, 29

## **J**

jobs

- listing scheduled, 16

- removing scheduled, 16

- scheduling to run later, 16

## **K**

key functions

- command line, 31

- InfoExplorer ASCII, 53

kill command, 17

Korn shell

- built-in commands, 38, 44

- command, 20, 38

- command aliasing, 41

- command grouping, 43

- command line editing, 47

- command substitution, 42

- comment, 38

- creating shell scripts, 40

- identifier, 38

- input and output, 43

- job control, 38



- list, 38
- metacharacters, 39
- parameter assignment list, 39
- parameter substitution, 42
- pattern matching, 42
- pipeline, 39
- quoting, 40
- reserved words, 41
- shell scripts, 40
- shell variable, 39
- simple command, 39
- starting, 40
- stopping, 40
- subshell, 39
- text editing commands, 48
- tilde substitution, 41
- vocabulary, 38
- wildcard character, 40
- word, 40

ksh command, 20

## **L**

ln command, 6

ls command, 6

## **M**

mail

- command, 17

- reading, 17

- sending, 17

mkdir command, 7

mount command, 26

mv command, 7

## **O**

online information

- accessing, 22

- HTML-based, 22

- InfoExplorer, 23

operating system, displaying name of, 19

## **P**

passwd command, 17

password, changing, 17

pathname, displaying current, 7



pg command, 1

printing

- canceling a print job, 14
- checking a print job, 13
- displaying print status, 13
- holding a print job, 14
- moving a print job, 15
- releasing a print job, 14
- starting a print job, 12

problems, solving, 28

processes

- listing, 16
- removing, 16
- scheduling jobs, 16
- showing current status, 18
- stopping, 17

ps command, 18

pwd command, 7

## Q

qcan command, 14

qchk command, 13

qhld command, 14

qmov command, 15

qprt command, 12

## R

remote host, accessing, 19

restore command, 25

rm command, 3

rmdir command, 8

## S

sar command, 29

scheduling jobs to run later, 16

sed command, 11

sh command, 20

shell startup commands, 20

shutdown, system, 18

shutdown command, 18

smit backup command, 24

SMIT command, 21

smit mount command, 27, 28

smit qcan command, 14

smit qchk command, 14



- smit qprt command, 13
- smit sar command, 30
- space, displaying information about, 28
- space utilization, analyzing, 28
- statistics
  - using iostat, 29
  - using vmstat, 30
- status of current processes, 18
- storage, writing files to or from, 26
- stty command, 19
- syntax diagrams, reading, vii
- system information, using sar, 29
- system management tasks, 50

## T

- tail command, 8
- tar command, 26
- time, displaying current, 22
- tn command, 19
- touch command, 10

## U

- umount command, 27
- uname command, 19
- uncompress command, 25
- users, identifying logged-in, 20

## V

- vi command, 12
- vi editor, 32
- vmstat command, 30

## W

- Web-based System Manager, 21, 50
- Web-based System Manager applications
  - backups, 51
  - devices, 51
  - file systems, 51
  - logical volumes, 51
  - network, 52
  - NIM, 52
  - printer queues, 52
  - processes, 52
  - software, 52
  - subsystems, 52





- system, 52
- users, 52
- who command, 20
- workstation parameters, setting, 19
- wsm backup command, 24, 51
- wsm devices command, 51
- wsm fs command, 27, 51
- wsm lvm command, 51
- wsm network command, 52
- wsm nim command, 52
- wsm printers command, 13, 14, 52
- wsm processes command, 52
- wsm software command, 52
- wsm subsystems command, 52
- wsm system command, 30, 52
- wsm users command, 52

L



**Vos remarques sur ce document**  
**Technical publication remark form**

<b>Titre / Title :</b> Bull Quick Reference Guide
--

<b>N° Référence / Reference N° :</b> 86 A2 55AP 02 <b>Daté / Dated :</b> June1998
--

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:

L

L



**Bull Electronics Europe S.A.**  
**Service CEDOC**  
331 Avenue Patton - BP 428  
49004 ANGERS CEDEX 01  
FRANCE

ORDER REFERENCE  
**86 A2 55AP 02**





Utiliser les marques de découpe pour  
obtenir les étiquettes.  
Use the cut marks to get the labels.



**Bull**  
AIX  
Quick Reference  
Guide

86 A2 55AP 02



**Bull**  
AIX  
Quick Reference  
Guide

86 A2 55AP 02

