

Virtual I/O Server and Integrated Virtualization Manager commands



ESCALA

Virtual I/O Server and Integrated Virtualization Manager commands

Hardware

May 2009

BULL CEDOC
357 AVENUE PATTON
B.P.20845
49008 ANGERS CEDEX 01
FRANCE

REFERENCE
86 A1 41EV 05

The following copyright notice protects this book under Copyright laws which prohibit such actions as, but not limited to, copying, distributing, modifying, and making derivative works.

Copyright © Bull SAS 2009

Printed in France

Trademarks and Acknowledgements

We acknowledge the rights of the proprietors of the trademarks mentioned in this manual.

All brand names and software and hardware product names are subject to trademark and/or patent protection.

Quoting of brand and product names is for information purposes only and does not represent trademark misuse.

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

Contents

Virtual I/O Server and Integrated Virtualization Manager commands listed alphabetically	1
Reading syntax diagrams	1
Exit status for Virtual I/O Server commands	1
activatevg command	2
alt_root_vg command	3
awk command	4
backup command	20
backupios command	25
IVM bkprofdata command	26
bootlist command	27
cat command	30
cattracerpt command	32
cfgassist command	32
cfgdev command	33
cfglnagg command	34
cfgnamesrv command	35
cfgsvc command	37
chbdsp command	42
chdate command	43
chdev command	44
IVM chhwres command	45
chlang command	52
IVM chled command	52
IVM chlparutil command	54
chlv command	55
chmod command	55
chpath command	59
chrep command	60
chsp command	61
IVM chsvcevent command	63
IVM chsyscfg command	64
IVM chsysstate command	72
chtcpip command	74
chuser command	75
IVM chvet command	77
chvg command	78
chvopt command	79
cl_snmp command	79
clear command	82
cp command	83
cplv command	86
cpvdi command	87
crontab command	88
date command	92
deactivatevg command	95
IVM defsysplanres command	96
IVM deploysysplan command	97
diagmenu command	99
dsmc command	100
entstat command	102
errlog command	106
exportvg command	106
extendlv command	107
extendvg command	108
fcstat command	109
fsck command	111

ftp command	112
grep command	124
head command	126
hostmap command	127
hostname command	129
importvg command	129
installios command	130
invscout command	132
ioslevel Command.	133
ldapadd command	133
ldapsearch command	134
ldfware command	136
license command	137
loadopt command	138
loginmsg command	138
IVM lpar_netboot command	139
IVM lpcfgop command	141
ls command	142
lsdev command	146
lsfailedlogin command	150
lsfware command	150
lsgcl command	151
IVM lshwres command	152
IVM lsled command	176
lslparinfo command	178
IVM lslparmigr command	178
IVM lslparutil command	181
lslv command	188
lsmap command	193
IVM lsmemdev command	198
lsnetvc command	200
lsnports command	201
lspath command	203
lspv command	206
IVM lsrefcode command	210
lsrep command	213
lssp command	214
lssvc command	216
IVM lssvcevents command	217
lssw command	227
IVM lssyscfg command	228
IVM lssysconn command	242
IVM lssysplan command	244
IVM lssysplanres command	245
lstcpip command	245
lsuser command	247
IVM lsvet command	248
lsvg command	249
lsvopt command	254
man command	255
migratepv command	255
IVM migrpar command	256
mirrorios command	260
IVM mkauthkeys command	261
mkbdspl command	263
mkdir command	264
IVM mkgencfg command	265
mkkrb5clnt command	268
mkldap command	268
mklv command	270
mklvcopy command	271

mkpath command	272
mkrep command	273
mksp command	274
IVM mksvcevent command	275
IVM mksyscfg command	276
IVM mksysplan command	284
mktcpip command	285
mkuser command	287
mkvdev command	288
mkvg command	291
mkvopt command	292
mkvt command	293
more command	293
motd command	298
mount command	298
mv command	299
netstat command	302
oem_platform_level command	306
oem_setup_env command	306
optimizenet command	307
IVM os_install Command	312
passwd command	316
pdump command	317
ping command	317
prepdev command	319
redefvg command	320
reducevg command	321
remote_management command	322
replphyvol command	323
restore command	324
restorevgstruct command	332
rm command	333
rmbdsp command	336
rmdev command	337
rmlv command	338
rmlvcopy command	338
rmpath command	339
rmrep command	341
rmsp command	341
IVM rmsyscfg command	342
IVM rmsysplan command	343
IVM rmsysplanres command	343
rmtcpip command	344
rmuser command	345
rmvdev command	346
rmvopt command	346
rmvt command	347
IVM rsthwres command	347
IVM rstprofdata command	350
save_base command	351
savevgstruct command	352
seastat command	353
sed command	354
showmount command	358
shutdown command	359
snap command	360
snmp_info command	361
snmp_trap command	363
snmpv3_ssw command	364
startnetsh command	364
startsvcs command	366

startsysdump command	367
starttrace command	368
stopnetsvc command	368
stopsvc command	370
stoptrace command	371
stty command	371
su command	378
svmon command	380
syncvg command	383
sysstat command	384
tail command	384
tee command	386
topas command	387
traceroute command	395
uname command	397
unloadopt command	399
unmirrorios command	399
unmount command	400
IVM update_install_setup command.	400
updateios command	402
vasistat command	403
vfcmap command	411
vi command.	412
viosecur command	431
viostat command	433
vmstat command	437
wall command	441
wc command	442
who command	444
wkldagent command	447
wkldmgr command	447
wkldout command	448

Appendix. Notices 451

Programming interface information	452
Trademarks	452
Terms and conditions.	453

Virtual I/O Server and Integrated Virtualization Manager commands listed alphabetically

This topic contains links to information about the Virtual I/O Server and Integrated Virtualization Manager commands, including syntax statements, flag descriptions, and usage examples. Commands are listed alphabetically.

Reading syntax diagrams

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

```
lsdev { -dev DeviceName | -plc PhysicalLocationCode } [ -child ] [ -field FieldName... ] [ -fmt Delimiter ]
```

The following conventions are used in the command syntax statements:

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

Exit status for Virtual I/O Server commands

The following table defines the standard return codes returned by all of the Virtual I/O Server commands. Additional return codes unique to a specific command are defined within the individual command description page.

Return Code	Description
0	Success
1	Syntax Error
2	Not Found
1	Command requires an option
1	Command requires the specified option
1	Command requires an attribute
3	Invalid access to execute command
1	Invalid command
1	Invalid flag or argument
1	Invalid option flag
1	Invalid attribute
1	Invalid option combination

Return Code	Description
1	Specified option requires an argument
1	Specified option requires an attribute
1	Specified option also requires another option
1	Specified option is repeated
1	Attributes cannot be repeated
1	Contains an invalid argument
1	Is invalid
1	Too many arguments
1	Too few arguments
1	Unable to acquire permission to execute command
4	Execution of this command did not complete
10	No device found with physical location
11	Too many matches for physical location
12	Too many matches for physical location code
18	Insufficient memory

activatevg command

Purpose

Activates a volume group.

Syntax

```
activatevg [-f ] VolumeGroup
```

Description

The **activatevg** command activates the volume group specified by the *VolumeGroup* parameter and all associated logical volumes. When a volume group is activated, physical partitions are synchronized if they are not current.

Flags

-f Allows a volume group to be made active that does not currently have a quorum of available disks.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

- To activate volume group **vg03**, type:

```
activatevg vg03
```

Related Information

The **mkvg** command, the **chvg** command, the **extendvg** command, the **reducevg** command, the **mirrorios** command, the **unmirrorios** command, the **lsvg** command, the **deactivatevg** command, the **importvg** command, the **exportvg** command, and the **syncvg** command.

alt_root_vg command

Purpose

Creates a copy of (clones) the currently running system to an alternative disk.

Syntax

```
alt_root_vg [-g][-z] [-bundle bundle_name -location images_location] -target target_disks...
```

Description

The **alt_root_vg** command allows users to copy the current root volume group to an alternate disk and to update the operating system to the next fix pack level, without taking the machine down for an extended period of time and mitigating outage risk. This can be done by creating a copy of the current rootvg on an alternate disk and simultaneously applying fix pack updates. If needed, the **bootlist** command can be run after the new disk has been booted, and the bootlist can be changed to boot back to the older level of the operating system.

Note: The *target_disk* cannot be a physical volume that is assigned to a shared memory pool (to be used as a paging space device by a shared memory partition).

Cloning the running rootvg, allows the user to create a backup copy of the root volume group. This copy can be used as a back up in case the rootvg failed, or it can be modified by installing additional updates. One scenario might be to clone a 1.3.0.0 system, and then install updates to bring the cloned rootvg to 1.3.0.0-FP8.0. This would update the system while it was still running. Rebooting from the new rootvg would bring the level of the running system to 1.3.0.0-FP8.0. If there was a problem with this level, changing the bootlist back to the 1.3.0.0 disk and rebooting would bring the system back to 1.3.0.0. Other scenarios would include cloning the rootvg and applying individual fixes, rebooting the system and testing those fixes, and rebooting back to the original rootvg if there was a problem.

At the end of the installation, a volume group, *altinst_rootvg*, is left on the target disks in the varied off state as a placeholder. If varied on, it indicates that it owns no logical volumes; however, the volume group does contain logical volumes, but they have been removed from the ODM because their names now conflict with the names of the logical volumes on the running system. Do not vary on the *altinst_rootvg* volume group; instead, leave the definition there as a placeholder.

After rebooting from the new alternate disk, the former rootvg volume group is displayed in a *lspv* listing as *old_rootvg*, and it includes all disks in the original rootvg. This former rootvg volume group is set to not vary-on at reboot.

If a return to the original rootvg is necessary, the *bootlist* command is used to reboot from the original rootvg.

Flags

-target *target_disks* Specifies a space-delimited list of the name or names of the target disks where the alternate rootvg will be created. These disks must not contain any volume group definitions. The **lspv** command shows these disks as belonging to volume group *None*.

-bundle <i>bundle_name</i>	Path name of optional file that contains a list of packages or file sets that are installed after a rootvg clone is created. You must also specify the -location flag when you specify the bundle option.
-location <i>image_location</i>	Location of the install images or updates to apply after a clone of rootvg is created. The location can be a directory and full-path name or device name, such as <i>/dev/rmt0</i> .
-g	Skips checks to ensure the disk can boot.
-z	Specifies that no types of non-rootvg volume groups are to be imported.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

To create a boot list of logical devices to be used in the next normal boot, enter the following commands:

```
alt_root_vg -target hdisk2 -bundle my_bundle -location /tmp/update
alt_root_vg -target hdisk2 -bundle my_bundle -location /home/padmin
```

Related Information

The **bootlist** command.

awk command

Purpose

Finds lines in files that match a pattern and performs specified actions on those lines.

Syntax

```
awk [ -F Ere ] [ -v Assignment ] ... { -f ProgramFile | 'Program' } [ [ File ... | Assignment ... ] ] ...
```

Description

The **awk** command utilizes a set of user-supplied instructions to compare a set of files, one line at a time, to extended regular expressions supplied by the user. Then actions are performed upon any line that matches the extended regular expressions.

The pattern searching of the **awk** command is more general than that of the **grep** command, and it allows the user to perform multiple actions on input text lines. The **awk** command programming language requires no compiling, and allows the user to use variables, numeric functions, string functions, and logical operators.

The **awk** command is affected by the **LANG**, **LC_ALL**, **LC_COLLATE**, **LC_CTYPE**, **LC_MESSAGES**, **LC_NUMERIC**, **NLSPATH**, and **PATH** environment variables.

The **awk** command takes two types of input: input text files and program instructions.

Input Text Files

Searching and actions are performed on input text files. The files are specified by:

- Specifying the *File* variable on the command line.
- Modifying the special variables **ARGV** and **ARGC**.
- Providing standard input in the absence of the *File* variable.

If multiple files are specified with the *File* variable, the files are processed in the order specified.

Program Instructions

Instructions provided by the user control the actions of the **awk** command. These instructions come from either the *Program* variable on the command line or from a file specified by the **-f** flag together with the *ProgramFile* variable. If multiple program files are specified, the files are concatenated in the order specified and the resultant order of instructions is used.

Output for the awk Command

The **awk** command produces three types of output from the data within the input text file:

- Selected data can be printed to standard output, without alteration to the input file.
- Selected portions of the input file can be altered.
- Selected data can be altered and printed to standard output, with or without altering the contents of the input file.

All of these types of output can be performed on the same file. The programming language recognized by the **awk** command allows the user to redirect output.

File Processing with Records and Fields

Files are processed in the following way:

1. The **awk** command scans its instructions and executes any actions specified to occur before the input file is read.

The **BEGIN** statement in the **awk** programming language allows the user to specify a set of instructions to be done before the first record is read. This is particularly useful for initializing special variables.

2. One record is read from the input file.

A record is a set of data separated by a record separator. The default value for the record separator is the new-line character, which makes each line in the file a separate record. The record separator can be changed by setting the **RS** special variable.

3. The record is compared against each pattern specified by the **awk** command's instructions.

The command instructions can specify that a specific field within the record be compared. By default, fields are separated by white space (blanks or tabs). Each field is referred to by a field variable. The first field in a record is assigned the **\$1** variable, the second field is assigned the **\$2** variable, and so forth. The entire record is assigned to the **\$0** variable. The field separator can be changed by using the **-F** flag on the command line or by setting the **FS** special variable. The **FS** special variable can be set to the values of: blank, single character, or extended regular expression.

4. If the record matches a pattern, any actions associated with that pattern are performed on the record.
5. After the record is compared to each pattern, and all specified actions are performed, the next record is read from input; the process is repeated until all records are read from the input file.
6. If multiple input files have been specified, the next file is then opened and the process repeated until all input files have been read.
7. After the last record in the last file is read, the **awk** command executes any instructions specified to occur after the input processing.

The **END** statement in the **awk** programming language allows the user to specify actions to be performed after the last record is read. This is particularly useful for sending messages about what work was accomplished by the **awk** command.

The awk Command Programming Language

The **awk** command programming language consists of statements in the form:

```
Pattern { Action }
```

If a record matches the specified pattern, or contains a field that matches the pattern, the associated action is then performed. A pattern can be specified without an action, in which case the entire line containing the pattern is written to standard output. An action specified without a pattern is performed for every input record.

Patterns

Four types of patterns are used in the **awk** command language syntax:

Regular Expressions

The extended regular expressions used by the **awk** command are similar to those used by the **grep** command. The simplest form of an extended regular expression is a string of characters enclosed in slashes. For an example, suppose a file named `testfile` had the following contents:

```
smawley, andy
smiley, allen
smith, alan
smithern, harry
smithhern, anne
smitters, alexis
```

Entering the following command line:

```
awk '/smi/' testfile
```

would print to standard output of all records that contained an occurrence of the string `smi`. In this example, the program `'/smi/'` for the **awk** command is a pattern with no action. The output is:

```
smiley, allen
smith, alan
smithern, harry
smithhern, anne
smitters, alexis
```

The following special characters are used to form extended regular expressions:

Character	Function
+	Specifies that a string matches if one or more occurrences of the character or extended regular expression that precedes the + (plus) are within the string. The command line: <pre>awk '/smith+ern/' testfile</pre> prints to standard output any record that contained a string with the characters <code>smi t</code> , followed by one or more <code>h</code> characters, and then ending with the characters <code>ern</code> . The output in this example is: <pre>smithern, harry smithhern, anne</pre>

Character	Function
?	<p>Specifies that a string matches if zero or one occurrences of the character or extended regular expression that precedes the ? (question mark) are within the string. The command line:</p> <pre>awk '/smith?/' testfile</pre> <p>prints to standard output of all records that contain the characters <code>smi t</code>, followed by zero or one instance of the <code>h</code> character. The output in this example is:</p> <pre>smith, alan smithern, harry smithhern, anne smitters, alexis</pre>
	<p>Specifies that a string matches if either of the strings separated by the (vertical line) are within the string. The command line:</p> <pre>awk '/allen alan /' testfile</pre> <p>prints to standard output of all records that contained the string <code>allen</code> or <code>alan</code>. The output in this example is:</p> <pre>smiley, allen smith, alan</pre>
()	<p>Groups strings together in regular expressions. The command line:</p> <pre>awk '/a(11)?(nn)?e/' testfile</pre> <p>prints to standard output of all records with the string <code>ae</code> or <code>alle</code> or <code>anne</code> or <code>allnne</code>. The output in this example is:</p> <pre>smiley, allen smithhern, anne</pre>
{ <i>m</i> }	<p>Specifies that a string matches if exactly <i>m</i> occurrences of the pattern are within the string. The command line:</p> <pre>awk '/l{2}/' testfile</pre> <p>prints to standard output</p> <pre>smiley, allen</pre>
{ <i>m</i> ,}	<p>Specifies that a string matches if at least <i>m</i> occurrences of the pattern are within the string. The command line:</p> <pre>awk '/t{2,}/' testfile</pre> <p>prints to standard output:</p> <pre>smitters, alexis</pre>
{ <i>m</i> , <i>n</i> }	<p>Specifies that a string matches if between <i>m</i> and <i>n</i>, inclusive, occurrences of the pattern are within the string (where $m \leq n$). The command line:</p> <pre>awk '/er{1, 2}/' testfile</pre> <p>prints to standard output:</p> <pre>smithern, harry smithern, anne smitters, alexis</pre>
[<i>String</i>]	<p>Signifies that the regular expression matches any characters specified by the <i>String</i> variable within the square brackets. The command line:</p> <pre>awk '/sm[a-h]/' testfile</pre> <p>prints to standard output of all records with the characters <code>sm</code> followed by any character in alphabetical order from <code>a</code> to <code>h</code>. The output in this example is:</p> <pre>smawley, andy</pre>

Character	Function
[<i>String</i>]	<p>A ^ (caret) within the [] (square brackets) and at the beginning of the specified string indicates that the regular expression <i>does not</i> match any characters within the square brackets. Thus, the command line:</p> <pre>awk '/sm[^a-h]/' testfile</pre> <p>prints to standard output:</p> <pre>smiley, allen smith, alan smithern, harry smithhern, anne smitters, alexis</pre>
~,!~	<p>Signifies a conditional statement that a specified variable matches (tilde) or does not match (tilde, exclamation point) the regular expression. The command line:</p> <pre>awk '\$1 ~ /n/' testfile</pre> <p>prints to standard output of all records whose first field contained the character n. The output in this example is:</p> <pre>smithern, harry smithhern, anne</pre>
^	<p>Signifies the beginning of a field or record. The command line:</p> <pre>awk '\$2 ~ /^h/' testfile</pre> <p>prints to standard output of all records with the character h as the first character of the second field. The output in this example is:</p> <pre>smithern, harry</pre>
\$	<p>Signifies the end of a field or record. The command line:</p> <pre>awk '\$2 ~ /y\$/' testfile</pre> <p>prints to standard output of all records with the character y as the last character of the second field. The output in this example is:</p> <pre>smawley, andy smithern, harry</pre>
. (period)	<p>Signifies any one character except the terminal new-line character at the end of a space. The command line:</p> <pre>awk '/a..e/' testfile</pre> <p>prints to standard output of all records with the characters a and e separated by two characters. The output in this example is:</p> <pre>smawley, andy smiley, allen smithhern, anne</pre>
*(asterisk)	<p>Signifies zero or more of any characters. The command line:</p> <pre>awk '/a.*e/' testfile</pre> <p>prints to standard output of all records with the characters a and e separated by zero or more characters. The output in this example is:</p> <pre>smawley, andy smiley, allen smithhern, anne smitters, alexis</pre>

Character	Function
\ (backslash)	<p>The escape character. When preceding any of the characters that have special meaning in extended regular expressions, the escape character removes any special meaning for the character. For example, the command line:</p> <pre>/a\\//</pre> <p>would match the pattern a //, since the backslashes negate the usual meaning of the slash as a delimiter of the regular expression. To specify the backslash itself as a character, use a double backslash. See the following item on escape sequences for more information on the backslash and its uses.</p>

Recognized Escape Sequences

The **awk** command recognizes most of the escape sequences used in C language conventions, as well as several that are used as special characters by the **awk** command itself. The escape sequences are:

Escape Sequence	Character Represented
\"	\" (double-quotation) mark
\/	/ (slash) character
\ddd	Character whose encoding is represented by a one-, two- or three-digit octal integer, where <i>d</i> represents an octal digit
\\	\ (backslash) character
\a	Alert character
\b	Backspace character
\f	Form-feed character
\n	New-line character (see following note)
\r	Carriage-return character
\t	Tab character
\v	Vertical tab.

Note: Except in the **gsub**, **match**, **split**, and **sub** built-in functions, the matching of extended regular expressions is based on input records. Record-separator characters (the new-line character by default) cannot be embedded in the expression, and no expression matches the record-separator character. If the record separator is not the new-line character, then the new-line character can be matched. In the four built-in functions specified, matching is based on text strings, and any character (including the record separator) can be embedded in the pattern so that the pattern matches the appropriate character. However, in all regular-expression matching with the **awk** command, the use of one or more NULL characters in the pattern produces undefined results.

Relational Expressions

The relational operators < (less than), > (greater than), <= (less than or equal to), >= (greater than or equal to), = (equal to), and != (not equal to) can be used to form patterns. For example, the pattern:

```
$1 < $4
```

matches records where the first field is less than the fourth field. The relational operators also work with string values. For example:

```
$1 != "q"
```

matches all records where the first field is not a q. String values can also be matched on collation values. For example:

```
$1 >= "d"
```

matches all records where the first field starts with a character that is a, b, c, or d. If no other information is given, field variables are compared as string values.

Combinations of Patterns

Patterns can be combined using three options:

- Ranges are specified by two patterns separated with a , (comma). Actions are performed on every record starting with the record that matches the first pattern, and continuing through and including the record that matches the second pattern. For example:

```
/begin/,/end/
```

matches the record containing the string begin, and every record between it and the record containing the string end, including the record containing the string end.

- Parentheses () group patterns together.
- The boolean operators || (or), && (and), and ! (not) combine patterns into expressions that match if they evaluate true, otherwise they do not match. For example, the pattern:

```
$1 == "a1" && $2 == "123"
```

matches records where the first field is a1 and the second field is 123.

BEGIN and END Patterns

Actions specified with the **BEGIN** pattern are performed before any input is read. Actions specified with the **END** pattern are performed after all input has been read. Multiple **BEGIN** and **END** patterns are allowed and processed in the order specified. An **END** pattern can precede a **BEGIN** pattern within the program statements. If a program consists only of **BEGIN** statements, the actions are performed and no input is read. If a program consists only of **END** statements, all the input is read prior to any actions being taken.

Actions

There are several types of action statements:

Action Statements

Action statements are enclosed in { } (braces). If the statements are specified without a pattern, they are performed on every record. Multiple actions can be specified within the braces, but must be separated by new-line characters or ; (semicolons), and the statements are processed in the order they appear. Action statements include:

Arithmetical Statements

The mathematical operators + (plus), - (minus), / (division), ^ (exponentiation), * (multiplication), % (modulus) are used in the form:

```
Expression Operator Expression
```

Thus, the statement:

```
$2 = $1 ^ 3
```

assigns the value of the first field raised to the third power to the second field.

Unary Statements

The unary - (minus) and unary + (plus) operate as in the C programming language:

```
+Expression or -Expression
```

Increment and Decrement Statements

The pre-increment and pre-decrement statements operate as in the C programming language:

```
++Variable or --Variable
```

The post-increment and post-decrement statements operate as in the C programming language:
Variable++ or Variable--

Assignment Statements

The assignment operators += (addition), -= (subtraction), /= (division), and *= (multiplication) operate as in the C programming language, with the form:

Variable += Expression

Variable -= Expression

Variable /= Expression

Variable *= Expression

For example, the statement:

```
$1 *= $2
```

multiplies the field variable **\$1** by the field variable **\$2** and then assigns the new value to **\$1**.

The assignment operators ^= (exponentiation) and %= (modulus) have the form:

Variable1^=Expression1

AND

Variable2%=Expression2

and they are equivalent to the C programming language statements:

```
Variable1=pow(Variable1, Expression1)
```

AND

```
Variable2=fmod(Variable2, Expression2)
```

where pow is the **pow** subroutine and fmod is the **fmod** subroutine.

String Concatenation Statements

String values can be concatenated by stating them side by side. For example:

```
$3 = $1 $2
```

assigns the concatenation of the strings in the field variables **\$1** and **\$2** to the field variable **\$3**.

Built-In Functions

The **awk** command language uses arithmetic functions, string functions, and general functions. The close Subroutine statement is necessary if you intend to write a file, then read it later in the same program.

Arithmetic Functions

The following arithmetic functions perform the same actions as the C language subroutines by the same name:

Function	Action
atan2 (<i>y</i> , <i>x</i>)	Returns arctangent of y/x .
cos (<i>x</i>)	Returns cosine of x ; x is in radians.
sin (<i>x</i>)	Returns sin of x ; x is in radians.
exp (<i>x</i>)	Returns the exponential function of x .
log (<i>x</i>)	Returns the natural logarithm of x .
sqrt (<i>x</i>)	Returns the square root of x .
int (<i>x</i>)	Returns the value of x truncated to an integer.
rand ()	Returns a random number n , with $0 \leq n < 1$.
srand ([<i>Expr</i>])	Sets the seed value for the rand function to the value of the <i>Expr</i> parameter, or use the time of day if the <i>Expr</i> parameter is omitted. The previous seed value is returned.

String Functions

The string functions are:

Function

gsub(*Ere*, *Repl*, [*In*])

sub(*Ere*, *Repl*, [*In*])

index(*String1*, *String2*)

length [(*String*)]

blength [(*String*)]

substr(*String*, *M*, [*N*])

match(*String*, *Ere*)

Action

Performs exactly as the **sub** function, except that all occurrences of the regular expression are replaced.

Replaces the first occurrence of the extended regular expression specified by the *Ere* parameter in the string specified by the *In* parameter with the string specified by the *Repl* parameter. The **sub** function returns the number of substitutions. An & (ampersand) appearing in the string specified by the *Repl* parameter is replaced by the string in the *In* parameter that matches the extended regular expression specified by the *Ere* parameter. If no *In* parameter is specified, the default value is the entire record (the **\$0** record variable).

Returns the position, numbering from 1, within the string specified by the *String1* parameter where the string specified by the *String2* parameter occurs. If the *String2* parameter does not occur in the *String1* parameter, a 0 (zero) is returned.

Returns the length, in characters, of the string specified by the *String* parameter. If no *String* parameter is given, the length of the entire record (the **\$0** record variable) is returned.

Returns the length, in bytes, of the string specified by the *String* parameter. If no *String* parameter is given, the length of the entire record (the **\$0** record variable) is returned.

Returns a substring with the number of characters specified by the *N* parameter. The substring is taken from the string specified by the *String* parameter, starting with the character in the position specified by the *M* parameter. The *M* parameter is specified with the first character in the *String* parameter as number 1. If the *N* parameter is not specified, the length of the substring will be from the position specified by the *M* parameter until the end of the *String* parameter.

Returns the position, in characters, numbering from 1, in the string specified by the *String* parameter where the extended regular expression specified by the *Ere* parameter occurs, or else returns a 0 (zero) if the *Ere* parameter does not occur. The **RSTART** special variable is set to the return value. The **RLENGTH** special variable is set to the length of the matched string, or to -1 (negative one) if no match is found.

Function

split(*String*, *A*, [*Ere*])

tolower(*String*)

toupper(*String*)

sprintf(*Format*, *Expr*, *Expr*, . . .)

Action

Splits the string specified by the *String* parameter into array elements *A*[1], *A*[2], . . . , *A*[*n*], and returns the value of the *n* variable. The separation is done with the extended regular expression specified by the *Ere* parameter or with the current field separator (the **FS** special variable) if the *Ere* parameter is not given. The elements in the *A* array are created with string values, unless context indicates a particular element should also have a numeric value.

Returns the string specified by the *String* parameter, with each uppercase character in the string changed to lowercase. The uppercase and lowercase mapping is defined by the **LC_CTYPE** category of the current locale.

Returns the string specified by the *String* parameter, with each lowercase character in the string changed to uppercase. The uppercase and lowercase mapping is defined by the **LC_CTYPE** category of the current locale.

Formats the expressions specified by the *Expr* parameters according to the **printf** subroutine format string specified by the *Format* parameter and returns the resulting string.

General Functions

The general functions are:

Function

close(*Expression*)

system(*Command*)

Action

Close the file or pipe opened by a **print** or **printf** statement or a call to the **getline** function with the same string-valued *Expression* parameter. If the file or pipe is successfully closed, a 0 is returned; otherwise a non-zero value is returned. The **close** statement is necessary if you intend to write a file, then read the file later in the same program.

Executes the command specified by the *Command* parameter and returns its exit status. Equivalent to the **system** subroutine.

Function

Expression | **getline** [*Variable*]

getline [*Variable*] < *Expression*

getline [*Variable*]

Action

Reads a record of input from a stream piped from the output of a command specified by the *Expression* parameter and assigns the value of the record to the variable specified by the *Variable* parameter. The stream is created if no stream is currently open with the value of the *Expression* parameter as its command name. The stream created is equivalent to one created by a call to the **popen** subroutine with the *Command* parameter taking the value of the *Expression* parameter and the *Mode* parameter set to a value of **r**. Each subsequent call to the **getline** function reads another record, as long as the stream remains open and the *Expression* parameter evaluates to the same string. If a *Variable* parameter is not specified, the **\$0** record variable and the **NF** special variable are set to the record read from the stream. Reads the next record of input from the file named by the *Expression* parameter and sets the variable specified by the *Variable* parameter to the value of the record. Each subsequent call to the **getline** function reads another record, as long as the stream remains open and the *Expression* parameter evaluates to the same string. If a *Variable* parameter is not specified, the **\$0** record variable and the **NF** special variable are set to the record read from the stream. Sets the variable specified by the *Variable* parameter to the next record of input from the current input file. If no *Variable* parameter is specified, **\$0** record variable is set to the value of the record, and the **NF**, **NR**, and **FNR** special variables are also set.

Note: All forms of the **getline** function return 1 for successful input, zero for end of file, and -1 for an error.

User-Defined Functions

User-defined functions are declared in the following form:

```
function Name (Parameter, Parameter,...) { Statements }
```

A function can be referred to anywhere in an **awk** command program, and its use can precede its definition. The scope of the function is global.

Function parameters can be either scalars or arrays. Parameter names are local to the function; all other variable names are global. The same name should not be used for different entities; for example, a parameter name should not be duplicated as a function name, or special variable. Variables with global scope should not share the name of a function. Scalars and arrays should not have the same name in the same scope.

The number of parameters in the function definition does not have to match the number of parameters used when the function is called. Excess formal parameters can be used as local variables. Extra scalar

parameters are initialized with a string value equivalent to the empty string and a numeric value of 0 (zero); extra array parameters are initialized as empty arrays.

When invoking a function, no white space is placed between the function name and the opening parenthesis. Function calls can be nested and recursive. Upon return from any nested or recursive function call, the values of all the calling function's parameters shall be unchanged, except for array parameters passed by reference. The **return** statement can be used to return a value.

Within a function definition, the new-line characters are optional before the opening { (brace) and after the closing } (brace).

An example of a function definition is:

```
function average ( g,n)
{
    for (i in g)
        sum=sum+g[i]
    avg=sum/n
    return avg
}
```

The function average is passed an array, g, and a variable, n, with the number of elements in the array. The function then obtains an average and returns it.

Conditional Statements

Most conditional statements in the **awk** command programming language have the same syntax and function as conditional statements in the C programming language. All of the conditional statements allow the use of { } (braces) to group together statements. An optional new-line can be used between the expression portion and the statement portion of the conditional statement, and new-lines or ; (semicolon) are used to separate multiple statements in { } (braces). Six conditional statements in C language are:

Conditional statement	Required syntax or description
if	if (<i>Expression</i>) { <i>Statement</i> } [else <i>Action</i>]
while	while (<i>Expression</i>) { <i>Statement</i> }
for	for (<i>Expression</i> ; <i>Expression</i> ; <i>Expression</i>) { <i>Statement</i> }
break	Causes the program loop to be exited when the break statement is used in either a while or for statement.
continue	Causes the program loop to move to the next iteration when the continue statement is used in either a while or for statement.

Five conditional statements in the **awk** command programming language that do not follow C-language rules are:

Conditional statement	Required syntax or description
for...in	for (<i>Variable in Array</i>) { <i>Statement</i> }

The **for...in** statement sets the *Variable* parameter to each index value of the *Array* variable, one index at a time and in no particular order, and performs the action specified by the *Statement* parameter with each iteration. See the **delete** statement for an example of a **for...in** statement.

Conditional statement	Required syntax or description
if...in	<p>if (<i>Variable in Array</i>) { <i>Statement</i> }</p> <p>The if...in statement searches for the existence of the <i>Array</i> element. The statement is performed if the <i>Array</i> element is found.</p>
delete	<p>delete <i>Array</i> [<i>Expression</i>]</p> <p>The delete statement deletes both the array element specified by the <i>Array</i> parameter and the index specified by the <i>Expression</i> parameter. For example, the statements:</p> <pre>for (i in g) delete g[i];</pre> <p>would delete every element of the <i>g</i>[] array.</p>
exit	<p>exit [<i>Expression</i>]</p> <p>The exit statement first invokes all END actions in the order they occur, then terminates the awk command with an exit status specified by the <i>Expression</i> parameter. No subsequent END actions are invoked if the exit statement occurs within an END action.</p>
#	<p># <i>Comment</i></p> <p>The # statement places comments. Comments should always end with a new-line but can begin anywhere on a line.</p>
next	Stops the processing of the current input record and proceeds with the next input record.

Output Statements

Two output statements in the **awk** command programming language are:

Output statement	Syntax and description
print	<p>print [<i>ExpressionList</i>] [<i>Redirection</i>] [<i>Expression</i>]</p> <p>The print statement writes the value of each expression specified by the <i>ExpressionList</i> parameter to standard output. Each expression is separated by the current value of the OFS special variable, and each record is terminated by the current value of the ORS special variable.</p> <p>The output can be redirected using the <i>Redirection</i> parameter, which can specify the three output redirections with the > (greater than), >> (double greater than), and the (pipe). The <i>Redirection</i> parameter specifies how the output is redirected, and the <i>Expression</i> parameter is either a path name to a file (when <i>Redirection</i> parameter is > or >>) or the name of a command (when the <i>Redirection</i> parameter is a).</p>
printf	<p>printf <i>Format</i> [, <i>ExpressionList</i>] [<i>Redirection</i>] [<i>Expression</i>]</p> <p>The printf statement writes to standard output the expressions specified by the <i>ExpressionList</i> parameter in the format specified by the <i>Format</i> parameter. The printf statement functions exactly like the printf command, except for the <i>c</i> conversion specification (%c). The <i>Redirection</i> and <i>Expression</i> parameters function the same as in the print statement.</p> <p>For the <i>c</i> conversion specification: if the argument has a numeric value, the character whose encoding is that value will be output. If the value is zero or is not the encoding of any character in the character set, the behavior is undefined. If the argument does not have a numeric value, the first character of the string value will be output; if the string does not contain any characters the behaviour is undefined.</p>

Note: If the *Expression* parameter specifies a path name for the *Redirection* parameter, the *Expression* parameter should be enclosed in double quotation marks to ensure that it is treated as a string.

Variables

Variables can be scalars, field variables, arrays, or special variables. Variable names cannot begin with a digit.

Variables can be used just by referencing them. With the exception of function parameters, they are not explicitly declared. Uninitialized scalar variables and array elements have both a numeric value of 0 (zero) and a string value of the null string ("").

Variables take on numeric or string values according to context. Each variable can have a numeric value, a string value, or both. For example:

```
x = "4" + "8"
```

assigns the value of 12 to the variable x. For string constants, expressions should be enclosed in "" (double quotation) marks.

There are no explicit conversions between numbers and strings. To force an expression to be treated as a number, add 0 (zero) to it. To force an expression to be treated as a string, append a null string ("").

Field Variables

Field variables are designated by a \$ (dollar sign) followed by a number or numerical expression. The first field in a record is assigned the \$1 variable, the second field is assigned to the \$2 variable, and so forth. The \$0 field variable is assigned to the entire record. New field variables can be created by assigning a value to them. Assigning a value to a non-existent field, that is, any field larger than the current value of \$NF field variable, forces the creation of any intervening fields (set to the null string), increases the value of the NF special variable, and forces the value of \$0 record variable to be recalculated. The new fields are separated by the current field separator (which is the value of the FS special variable). Blanks and tabs are the default field separators. To change the field separator, use the -F flag, or assign the FS special variable a different value in the **awk** command program.

Arrays

Arrays are initially empty and their sizes change dynamically. Arrays are represented by a variable with subscripts in [] (square brackets). The subscripts, or element identifiers, can be numbers or strings, which provide a type of associative array capability. For example, the program:

```
/red/ { x["red"]++ }  
/green/ { y["green"]++ }
```

increments counts for both the red counter and the green counter.

Arrays can be indexed with more than one subscript, similar to multidimensional arrays in some programming languages. Because programming arrays for the **awk** command are really one dimensional, the comma-separated subscripts are converted to a single string by concatenating the string values of the separate expressions, with each expression separated by the value of the **SUBSEP** environmental variable. Therefore, the following two index operations are equivalent:

```
x[expr1, expr2, ...exprn]
```

AND

```
x[expr1SUBSEPexpr2SUBSEP...SUBSEPexprn]
```

When using the **in** operator, a multidimensional *Index* value should be contained within parentheses. Except for the **in** operator, any reference to a nonexistent array element automatically creates that element.

Special Variables

The following variables have special meaning for the **awk** command:

Special variable	Description
ARGC	The number of elements in the ARGV array. This value can be altered.
ARGV	The array with each member containing one of the <i>File</i> variables or <i>Assignment</i> variables, taken in order from the command line, and numbered from 0 (zero) to ARGC -1. As each input file is finished, the next member of the ARGV array provides the name of the next input file, unless: <ul style="list-style-type: none"> • The next member is an <i>Assignment</i> statement, in which case the assignment is evaluated. • The next member has a null value, in which case the member is skipped. Programs can skip selected input files by setting the member of the ARGV array that contains that input file to a null value. • The next member is the current value of ARGV [ARGC -1], which the awk command interprets as the end of the input files.
CONVFMT	The printf format for converting numbers to strings (except for output statements, where the OFMT special variable is used). The default is "%.6g".
ENVIRON	An array representing the environment under which the awk command operates. Each element of the array is of the form: <p>ENVIRON ["<i>Environment VariableName</i>"] = <i>EnvironmentVariableValue</i></p> <p>The values are set when the awk command begins execution, and that environment is used until the end of execution, regardless of any modification of the ENVIRON special variable.</p>
FILENAME	The path name of the current input file. During the execution of a BEGIN action, the value of FILENAME is undefined. During the execution of an END action, the value is the name of the last input file processed.
FNR	The number of the current input record in the current file.
FS	The input field separator. The default value is a blank. If the input field separator is a blank, any number of locale-defined spaces can separate fields. The FS special variable can take two additional values: <ul style="list-style-type: none"> • With FS set to a single character, fields are separated by each single occurrence of the character. • With FS set to an extended regular expression, each occurrence of a sequence matching the extended regular expression separates fields.
NF	The number of fields in the current record, with a limit of 99. Inside a BEGIN action, the NF special variable is undefined unless a getline function without a <i>Variable</i> parameter has been issued previously. Inside an END action, the NF special variable retains the value it had for the last record read, unless a subsequent, redirected, getline function without a <i>Variable</i> parameter is issued prior to entering the END action.
NR	The number of the current input record. Inside a BEGIN action the value of the NR special variable is 0 (zero). Inside an END action, the value is the number of the last record processed.
OFMT	The printf format for converting numbers to strings in output statements. The default is "%.6g".
OFS	The output field separator (default is a space).
ORS	The output record separator (default is a new-line character).
RLENGTH	The length of the string matched by the match function.
RS	Input record separator (default is a new-line character). If the RS special variable is null, records are separated by sequences of one or more blank lines; leading or trailing blank lines do not result in empty records at the beginning or end of input; and the new-line character is always a field separator, regardless of the value of the FS special variable.
RSTART	The starting position of the string matched by the match function, numbering from 1. Equivalent to the return value of the match function.
SUBSEP	Separates multiple subscripts. The default is \031.

Flags

<code>-f ProgramFile</code>	Obtains instructions for the awk command from the file specified by the <i>ProgramFile</i> variable. If the <code>-f</code> flag is specified multiple times, the concatenation of the files, in the order specified, will be used as the set of instructions.
<code>-F Ere</code>	Uses the extended regular expression specified by the <i>Ere</i> variable as the field separator. The default field separator is a blank.
<code>-v Assignment</code>	Assigns a value to a variable for the awk command's programming language. The <i>Assignment</i> parameter is in the form of <i>Name = Value</i> . The <i>Name</i> portion specifies the name of the variable and can be any combination of underscores, digits, and alphabetic characters, but it must start with either an alphabetic character or an underscore. The <i>Value</i> portion is also composed of underscores, digits, and alphabetic characters, and is treated as if it were preceded and followed by a " (double-quotation character, similar to a string value). If the <i>Value</i> portion is numeric, the variable will also be assigned the numeric value.
<i>Assignment</i>	The assignment specified by the <code>-v</code> flag occurs before any portion of the awk command's program is executed, including the BEGIN section. Assigns a value to a variable for the awk command's programming language. It has the same form and function as the <i>Assignment</i> variable with the <code>-v</code> flag, except for the time each is processed. The <i>Assignment</i> parameter is processed just prior to the input file (specified by the <i>File</i> variable) that follows it on the command line. If the <i>Assignment</i> parameter is specified just prior to the first of multiple input files, the assignments are processed just after the BEGIN sections (if any). If an <i>Assignment</i> parameter occurs after the last file, the assignment is processed before the END sections (if any). If no input files are specified, the assignments are processed the standard input is read.
<i>File</i>	Specifies the name of the file that contains the input for processing. If no <i>File</i> variable is specified, or if a - (minus) sign is specified, standard input is processed.
<i>'Program'</i>	Contains the instructions for the awk command. If the <code>-f</code> flag is not specified, the <i>Program</i> variable should be the first item on the command line. It should be bracketed by ' ' (single quotation mark).

Exit Status

This command returns the following exit values:

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

Examples

1. To display the lines of a file that are longer than 72 characters, enter:

```
awk 'length >72' chapter1
```

This selects each line of the *chapter1* file that is longer than 72 characters and writes these lines to standard output, because no *Action* is specified. A tab character is counted as 1 byte.

2. To display all lines between the words *start* and *stop*, including "start" and "stop", enter:

```
awk '/start/,/stop/' chapter1
```

3. To run an **awk** command program, *sum2.awk*, that processes the file, *chapter1*, enter:

```
awk -f sum2.awk chapter1
```

The following program, *sum2.awk*, computes the sum and average of the numbers in the second column of the input file, *chapter1*:

```
{
    sum += $2
}
```

```

END {
    print "Sum: ", sum;
    print "Average:", sum/NR;
}

```

The first action adds the value of the second field of each line to the variable `sum`. All variables are initialized to the numeric value of 0 (zero) when first referenced. The pattern `END` before the second action causes those actions to be performed after all of the input file has been read. The `NR` special variable, which is used to calculate the average, is a special variable specifying the number of records that have been read.

4. To print the first two fields in opposite order, enter:

```
awk '{ print $2, $1 }' chapter1
```

5. The following `awk` program

```
awk -f sum3.awk chapter2
```

prints the first two fields of the file `chapter2` with input fields separated by comma and blanks and tabs, and then adds up the first column, and prints the sum and average:

```

BEGIN {FS = ",|[\t]+"}
      {print $1, $2}
      {s += $1}
END   {print "sum is",s,"average is", s/NR }

```

Related Information

The `grep` command and the `sed` command.

backup command

To back up a file system by *i-node*, specify the `-Level` and `FileSystem` parameters. When used in conjunction with the `-u` flag, the `-Level` parameter provides a method of maintaining a hierarchy of incremental backups for each file system. Specify the `-u` flag and set the `-Level` parameter to `n` to back up only those files that have been modified since the `n-1` level backup. The possible backup levels are 0 to 9. A level 0 backup archives all files in the file system.

The `FileSystem` parameter can specify either the physical device name (block or raw name) or the name of the directory on which the file system is mounted. The default file system is the root (`/`) file system.

Users must have read access to the file system device (such as `/dev/hd4`) or have backup authorization to perform backups by `i_node`.

Note:

1. You must first unmount a file system before backing it up by *i-node*. If you attempt to back up a mounted file system, a warning message is displayed. The `backup` command continues, but the created backup might contain inconsistencies because of changes that might have occurred in the file system during the backup operation.
2. Backing up file systems by *i-node* truncates the `uid` or `gid` of files having a `uid` or `gid` greater than 65535. When restored, these files might have different values for the `uid` and `gid` attributes. To retain the values correctly, always back up by name those files that have a `uid` or `gid` greater than 65535.
3. You can archive only JFS (Journaled File System) file systems when backing up by *i-node*. Back up any non-JFS file systems by file name.

Purpose

Backs up files and file systems.

Syntax

To back up files by name:

```
backup -i [ -b Number ] [ -p [ -e RegularExpression ] ] [ -E{force|ignore|warn} ] [ -f Device ] [ -l Number ]  
[ -U ] [ -o ] [ -q ] [ -v ]
```

To back up file systems by i-node:

```
backup [ [ -Level ] [ -b Number ] [ -c ] [ -f Device ] [ -L Length ] [ -U ] [ -u ] ] [ FileSystem ] | [ -w | -W ]
```

Description

The **backup** command creates copies of your files on a backup medium, such as a magnetic tape or diskette. The copies are in one of the two backup formats:

- Specific files backed up by name using the **-i** flag.
- Entire file system backed up by i-node using the *Level* and *FileSystem* parameters.

If you issue the **backup** command without any parameters, the root file system level 9 i-node back ups to the `/dev/rfd0` device. The default syntax is:

```
-9uf/dev/rfd0 /dev/rhd4
```

The default backup device is `/dev/rfd0`. If flags are specified that are not appropriate for the specified backup device, the **backup** command displays an error message and continues with the backup.

A single backup can span multiple volumes.

Note:

1. Running the **backup** command results in the loss of all material previously stored on the selected output medium.
2. Data integrity of the archive might be compromised if a file is modified during system backup. Keep system activity at a minimum during the system backup procedure.
3. If a backup is made to a tape device with the device block size set to 0, it might be difficult to restore data from the tape unless the default write size was used with the **backup** command. The default write size for the **backup** command can be read by the **restore** command when the tape device block size is 0.

In other words, the **-b** flag should not be specified when the tape device block size is 0. If the **-b** flag of the **backup** command is specified and is different from the default size, the same size must be specified with the **-b** flag of the **restore** command when the archived files are restored from the tape.

Backing up files by name

To back up by name, use the **-i** flag. The **backup** command reads standard input for the names of the files to be backed up.

File types can be special files, regular files, or directories. When the file type is a directory, only the directory is backed up. The files under the directory are not backed up, unless they are explicitly specified.

Note:

1. Files are restored using the same path names as the archived files. Therefore, to create a backup that can be restored from any path, use full path names for the files that you want to back up.
2. When backing up files that require multiple volumes, do not enter the list of file names from the keyboard. Instead, pipe or redirect the list from a file to the **backup** command.

When you enter the file names from the keyboard and the backup process needs a new tape or diskette, the command loses any file names already entered but not yet backed up. To avoid this problem, enter each file name only after the archived message for the previous file has been displayed. The archived message consists of the character a followed by the file name.

3. If you specify the **-p** flag, only files of less than 2 GB are packed.

Backing up file systems by i-node

Flags

-b *Number*

For backups by name, specifies the number of 512-byte blocks; for backups by i-node, specifies the number of 1024-byte blocks to write in a single output operation. When the **backup** command writes to tape devices, the default is 100 for backups by name and 32 for backups by i-node.

The write size is the number of blocks multiplied by the block size. The default write size for the **backup** command writing to tape devices is 51200 (100 × 512) for backups by name and 32768 (32 × 1024) for backups by i-node. The write size must be an even multiple of the tape's physical block size.

The value of the **-b** flag is always ignored when the **backup** command writes to diskette. In this case, the command always writes in clusters that occupy a complete track.

-c

Specifies that the tape is a cartridge, not a nine-track.

-e *RegularExpression*

Specifies that the files with names matching the regular expression are not to be packed. A regular expression is a set of characters, meta characters, and operators that define a string or group of strings in a search pattern. It can also be a string containing wildcard characters and operations that define a set of one or more possible strings.

The **-e** flag is applied only when the **-p** flag is specified.

-E

For backups by name, the **-E** flag requires one of the following arguments. If you omit the **-E** flag, **warn** is the default behavior.

force Fails the backup operation on a file if the fixed extent size or space reservation of the file cannot be preserved.

ignore Ignores any errors in preserving extent attributes.

warn Issues a warning if the space reservation or the fixed extent size of the file cannot be preserved.

-f Device

Specifies the output device. To send output to a named device, specify the *Device* variable as a path name such as the following example:

```
/dev/rmt0
```

To send output to the standard output device, specify a - (minus sign). The - (minus sign) feature enables you to pipe the output of the **backup** command to the **dd** command.

You can also specify a range of archive devices. The range specification must be in the following format:

```
/dev/deviceXXX-YYY
```

where XXX and YYY are whole numbers, and XXX must always be less than YYY; for example:

```
/dev/rfd0-3
```

All devices in the specified range must be of the same type. For example, you can use a set of 8 mm, 2.3 GB tapes or a set of 1.44 MB diskettes. All tape devices must be set to the same physical tape block size.

If the *Device* variable specifies a range, the **backup** command automatically goes from one device in the range to the next. After exhausting all of the specified devices, the **backup** command stops and requests that new volumes be mounted on the range of devices.

i

Specifies that files be read from standard input and archived by file name. If relative path names are used, files are restored (with the **restore** command) relative to the current directory at restore time. If full path names are used, files are restored to those same names.

-L 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 MB), respectively. To represent a tape length of 2 GB, enter **-L 2g**.

Note: Use the **-L** flag only for i-node backups.

-l Number

Limits the total number of blocks to use on the diskette device. The value specified must be a nonzero multiple of the number of sectors per diskette track. This option applies to by-name backups only.

-o

Creates a Version 2-compatible backup by name. This flag is required for compatibility with Version 2 systems because backups by name that are created by a version higher than 2 cannot be restored on Version 2 systems. To create a Version 2-compatible backup by name, use the **-o** flag along with other flags required for backups by name.

Files with attributes and values, such as user IDs and group IDs, that are too large for Version 2 systems are not backed up. A message is displayed for each such file and each value that is too large.

-p

Specifies that the files be packed, or compressed, before they are archived. Only files smaller than 2 GB are packed.

Note: This option should only be used when backing up files from an inactive file system. Modifying a file when a backup is in progress might result in corruption of the backup and an inability to recover the data. When backing up to a tape device that performs compression, this option can be omitted.

-q

Indicates that the removable medium is ready to use. When you specify the **-q** flag, the backup command proceeds without prompting you to prepare the backup medium. Press the Enter key to continue. This option applies only to the first volume. You are prompted for subsequent volumes. The **-q** flag applies only to backups by name.

-U

Specifies to backup any Access Control List (ACL)s or named extended attributes. Without this option the image will include only AIXC ACLs and Printing Color Files (PCL)s in the archive along with the other regular file data. For files containing NFS4 ACLs, conversion to AIXC will happen by default during archival.

-u	Updates the <code>/etc/dumpdates</code> file with the raw device name of the file system and the time, date, and level of the backup. You must specify the -u flag if you are making incremental backups. The -u flag applies only to backups by i-node.
-v	Causes the backup command to display additional information about the backup. When using the -v flag, the size of the file as it exists on the archive is displayed in bytes. Additionally, a total of these file sizes is displayed when all files have been processed. Directories are listed with a size of 0. Symbolic links are listed with the size of the symbolic link. Hard links are listed with the size of the file, which is how hard links are archived. Block and character devices, if they were backed up, are listed with a size of 0. When the -v flag is not specified, the backup command displays only the names of the files being archived. This option is used only when backing up by file name.
-w	Disabled. If the -w flag is specified, no other flags are applied.
-W	Displays, for each file system in the <code>/etc/dumpdates</code> file, the most recent backup date and level. If the -W flag is specified, no other flags are applied.
-Level	Specifies the backup level (0 to 9). The default level is 9.

Exit Status

This command returns the following exit values:

0	Successful completion.
>0	An error occurred.

Examples

1. To back up all the files and subdirectories in the `/home` directory using full path names, type the following command:

```
find /home -print | backup -i -f /dev/rmt0
```

The **-i** flag specifies that files will be read from standard input and archived by the file name. The **find** command generates a list of all the files in the `/home` directory. The files in this list are full path names. The `|` (pipe symbol) causes this list to be read from standard input by the **backup** command. The **-f** flag directs the **backup** command to write the files to the `/dev/rmt0` tape device. Because the files are archived using full path names, they will be written to the same paths when restored.

2. To back up all the files and subdirectories in the `/home/padmin` directory using relative path names, type the following command:

```
find . -print | backup -i -v -q
```

Each file name in the list generated by the **find** command is preceded by `./` (dot, slash). Because the files are backed up using relative path names, they are written to the current directory when restored. The **-v** flag causes the **backup** command to display additional information about the backup. The files are written to the default backup device `/dev/rfd0`.

3. To back up the `/` (root) file system, type the following command:

```
backup -0 -u -f /dev/rmt0 /
```

The 0 level specifies that all the files in the `/` (root) file system be backed up. The **-u** flag causes the backup command to update the `/etc/dumpdates` file for this backup.

4. To back up all the files in the root (`/`) file system that have been modified since the last level 0 backup, type the following command:

```
backup -1 -u -f /dev/rmt0 /
```

If the `/etc/dumpdates` file does not have an entry for a level 0 backup of the `/` (root) system, all the files in the file system are backed up.

5. To back up virtual optical media files from the virtual media repository, complete the following tasks:
 - a. Create a file called `backup_files` with the names of the files to be backed up. Include the full path name.

- ```
/var/vio/VMLibrary/media_file1
/var/vio/VMLibrary/media_file2
/var/vio/VMLibrary/media_filen
```
- b. Send the file using the **backup** command:
- ```
cat backup_files | backup -i -f /dev/rmt0
```

Related Information

The **restore** command.

backupios command

Purpose

Creates an installable image of the root volume group either onto a bootable tape, file system, or DVD.

Syntax

```
backupios -file {Directory} [-nosvg]
```

```
backupios -file {File name} [-mkysb] [-nosvg]
```

```
backupios -tape Device [-nopak] [-verify] [-nosvg]
```

```
backupios -cd Device {-udf | -cdformat} [-accept] [-nosvg]
```

Description

The **backupios** command creates a backup of the Virtual I/O Server and places it onto a file system, bootable tape, or DVD. You can use this backup to reinstall a system to its original state after it has been corrupted. If you create the backup on tape, the tape is bootable and includes the installation programs needed to install from the backup.

If the **-cd** flag is specified, the **backupios** command creates a system backup image to DVD-RAM media. If you need to create multi-volume discs because the image does not fit on one disc, the **backupios** command gives instructions for disk replacement and removal until all the volumes have been created.

Note: Vendor disc drives may support burning to additional disc types, such as CD-RW and DVD-R. Refer to the documentation for your drive to determine which disc types are supported.

If the **-file** flag is specified, the **backupios** command creates a system backup image to the path specified. The file system must be mounted and writable by the Virtual I/O Server root user prior to running the **backupios** command (see “mount command” on page 298 for details). Backing up the Virtual I/O Server to a remote file system will create the **nim_resources.tar** image in the directory you specify. The Virtual I/O Server must have root write access to the server on which the backup will be created. This backup can be reinstalled from the HMC using the **installios** command.

The **backupios** command empties the **target_disks_stanza** section of **bosinst.data** (which is part of the **nim_resources.tar** image) and sets **RECOVER_DEVICES=Default**. This allows the **mkysb** file generated by the command to be cloned to another logical partition. If you plan to use the **nim_resources.tar** image to install to a specific disk, then you need to repopulate the **target_disks_stanza** section of **bosinst.data** and replace this file in the **nim_resources.tar** image. All other parts of the **nim_resources.tar** image must remain unchanged.

Prior to backing up the root volume group, the **backupios** command saves the structure of all user-defined volume groups by calling the **savevgstruct** command for each volume group defined. To avoid having all user-defined volume groups backed up, use the **-nosvg** flag.

Note: The **backupios** command backs up only the volume group structures that are activated. The structures of volumes groups that are deactivated are not backed up.

Flags

-accept	Accepts licenses.
-cd	The Virtual I/O Server backup is placed onto DVD-RAM media.
-cdformat	Creates final CD images that are DVD sized (up to 4.38 GB).
-file	The Virtual I/O Server backup is placed in a file.
-mkysb	When the -mkysb flag is used, the resources used by the installios command are not saved in the image.
-nopak	Disables software packing of the files as they are backed up. Some tape drives use their own packing or compression algorithms.
-nosvg	Prevents the volume groups structure of user defined volume groups from being saved as part of the backupios process.
-tape	The Virtual I/O Server backup is placed onto a tape.
-udf	Creates a UDF (Universal Disk Format) file system on DVD-RAM media. The default format is Rock Ridge (ISO9660).
-verify	Verifies a tape backup. This flag causes the backupios command to verify the file header of each file on the backup tape and report any read errors as they occur.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

1. To generate a backup to the optical device named **/dev/cd1**, type:

```
backupios -cd /dev/cd1 -cdformat
```
2. To generate a system backup to a tape device named **/dev/rmt0**, type:

```
backupios -tape /dev/rmt0
```

Related Information

The mount command and the savevgstruct command.

IVM bkprofdata command

Purpose

Backs up profile data. This command is operable only in an Integrated Virtualization Manager environment.

Syntax

```
bkprofdata -o backup [ -f BackupFile ] [ -m ManagedSystem ]
```

Description

The **bkprofdata** command performs a back up of logical partition configuration information to a file. The file can later be used to restore the partition configuration with the **rstprofdata** command.

Flags

-f <i>BackupFile</i>	The name of the file to write the dump to in the current working directory. If not specified, the default file is /var/adm/lpm/profile.bak .
-o <i>backup</i>	The backup parameter backs up logical partition configuration data to a file.
-m <i>ManagedSystem</i>	The name of the managed system. This attribute is optional because there is only one system to manage. The name may either be the user-defined name for the managed system, or be in the form <code>ttt-mmm*sssssss</code> , where <code>ttt</code> is the machine type, <code>mmm</code> is the model, and <code>sssssss</code> is the serial number of the managed system.

Exit Status

This command has a return code of 0 on success.

Security

This command is available to all users.

Examples

1. To back up the partition configuration data to **/var/adm/lpm/profile.bak**, type:
`bkprofdata -o backup`
2. To back up the partition configuration data to **lparData.bak**, type:
`bkprofdata -o backup -f lparData.bak`

Related Information

The `rstprofdata` command.

bootlist command

Purpose

Displays and alters the list of boot devices available to the system.

Syntax

```
bootlist -mode mode {[ -ls ] [ [ -rm ] | [ Device [ -attr Attribute=Value ... ] ... ] ] }
```

Description

The **bootlist** command allows the user to display and alter the list of possible boot devices from which the system may be booted. When the system is booted, it will scan the devices in the list and attempt to boot from the first device it finds containing a boot image. This command supports the updating of the following:

- Service boot list. The service list designates possible boot devices for when the system is booted in service mode. How a system is booted in service mode is hardware-platform dependent. It may require a key switch to be turned to the Service position, a particular function key to be pressed during the boot process, or some other mechanism, as defined for the particular hardware platform.
- Previous boot device entry. This entry designates the last device from which the system booted. Some hardware platforms may attempt to boot from the previous boot device before looking for a boot device in one of the other lists.

The **bootlist** command supports the specification of generic device types as well as specific devices for boot candidates. Devices in the boot device list occur in the same order as devices listed on the invocation of this command.

The selection of the boot list to display or alter is made with the **-mode** *mode* option, where the *mode* variable is one of the keywords: **service**, **normal**, **both**, or **prevboot**. If the **both** keyword is specified, then both the normal boot list and the service boot list will be displayed, or if being altered, will be set to the same list of devices. If the **prevboot** keyword is specified, the only alteration allowed is with the **-rm** flag. The **-rm** flag invalidates the boot list specified by the **-mode** flag.

The devices currently in the boot list may be displayed by using the **-ls** flag. The list of devices that make up the specified boot list will be displayed, one device per line. If a device specified in the boot list is no longer present on the system, a '-' is displayed instead of a name.

Note: When you add a hot plug adapter to the system, that adapter and its child devices might not be available for specification as a boot device when you use the **bootlist** command. You may be required to reboot your system to make all potential boot devices known to the operating system.

Device Choices

The device name specified on the command line can occur in one of two different forms:

- It can indicate a specific device by its device logical name.
- It can indicate a generic or special device type by keyword. The following generic device keywords are supported:

fd	Any standard I/O-attached diskette drive
scdisk	Any SCSI-attached disk (including serial-link disk drives)
badisk	Any direct bus-attached disk
cd	Any SCSI-attached CD-ROM
rmt	Any SCSI-attached tape device
ent	Any Ethernet adapter
fdi	Any Fiber Distributed Data Interface adapter

When a specific device is to be included in the device list, the device's logical name (used with system management commands) must be specified. This logical name is made up of a prefix and a suffix. The suffix is generally a number and designates the specific device. The specified device must be in the Available state. If it is not, the update to the device list is rejected and this command fails. The following devices and their associated logical names are supported (where the bold type is the prefix and the *xx* variable is the device-specific suffix):

fdxx	Diskette-drive device logical names
hdiskxx	Physical-volume device logical names
cdxx	SCSI CD-ROM device logical names
rmtxx	Magnetic-tape device logical names
entxx	Ethernet-adapter logical names
fdiixx	Fiber Distributed Data Interface adapter logical names

Attribute Choices

Attributes are extra pieces of information about a device that the user supplies on the command line. Since this information is specific to a particular device, generic devices do not have attributes. Attributes apply to the device that immediately precedes them on the command line, which allows attributes to be interspersed among devices on the command line. Currently, only network devices have attributes. These are:

bserver	IP address of the BOOTP server
gateway	IP address of the gateway
client	IP address of the client
speed	Network adapter speed
duplex	The mode of the network adapter

Error Handling

If this command returns with an error, the device lists are not altered. The following device list errors are possible:

- If the user attempts to display or alter a boot list that is not supported by the hardware platform, the command fails, indicating the mode is not supported.
- If the user attempts to add too many devices to the boot list, the command fails, indicating that too many devices were requested. The number of devices supported varies depending on the device selection and the hardware platform .
- If an invalid keyword, invalid flag, or unknown device is specified, the command fails with the appropriate error message.
- If a specified device is not in the Available state, the command fails with the appropriate error message.

Flags

<i>Device</i>	Provides the names of the specific or generic devices to include in the boot list.
-attr <i>Attribute=Value</i>	Specifies the device attribute value pairs to be used instead of the defaults. The <i>Attribute=Value</i> variable can be used to specify one attribute value pair or multiple attribute value pairs for one -attr flag. If you use an -attr flag with multiple attribute value pairs, the list of pairs must be enclosed in quotation marks with a blank space between the pairs. For example, entering -attr Attribute=Value lists one attribute value pair per flag, while entering -attr 'Attribute1=Value1 Attribute2=Value2' lists more than one attribute value pair.
-mode <i>Mode</i>	Specifies which boot list to display or alter. Possible values for the <i>mode</i> variable are normal , service , both , or prevboot .
-ls	Indicates that the specified boot list is to be displayed after any specified alteration is performed. The output is a list of device names.
-rm	Indicates that the device list specified by the -mode flag should be invalidated.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

1. To invalidate the Service mode boot list, type:

```
bootlist -mode service -rm
```
2. To make a boot list for Normal mode with devices listed on the command line, type:

```
bootlist -mode normal hdisk0 hdisk1 rmt0 fd
```
3. To attempt to boot through a gateway using Ethernet, and then try other devices, type a command similar to the following. Substitute the IP addresses specified in the example with your IP addresses.

```
bootlist -mode normal ent0 -attr gateway=129.35.21.1 bserver=129.12.2.10 \  
client=129.35.9.23 hdisk0 rmt0 tok0 bserver=129.35.10.19 hdisk1
```

Related Information

The backupios command.

cat command

Purpose

Concatenates or displays files.

Syntax

```
cat [ -q ] [ -r ] [ -s ] [ -S ] [ -u ] [ -n [ -b ] ] [ -v [ -e ] [ -t ] ] [ - | File ... ]
```

Description

The **cat** command reads each *File* parameter in sequence and writes it to standard output. If you do not specify a file name, the **cat** command reads from standard input. You can also specify a file name of - (dash) for standard input.

Attention: Do not redirect output to one of the input files using the redirection symbol, > (caret). If you do this, you lose the original data in the input file because the shell truncates the file before the **cat** command can read it.

Flags

-b	Omits line numbers from blank lines, when specified with the -n flag.
-e	Displays a \$ (dollar sign) at the end of each line, when specified with the -v flag.
-n	Displays output lines preceded by line numbers, numbered sequentially from 1.
-q	Does not display a message if the cat command cannot find an input file. This flag is identical to the -s flag.
-r	Replaces multiple consecutive empty lines with one empty line. This flag is identical to the -S flag.
-s	Does not display a message if the cat command cannot find an input file. This flag is identical to the -q flag.
-S	Replaces multiple consecutive empty lines with one empty line. This flag is identical to the -r flag.
-t	Displays tab characters as ^I if specified with the -v flag.
-u	Does not buffer output. The default is buffered output.
-v	Displays nonprinting characters as visible characters, with the exception of tabs, new-lines, and form-feeds. ASCII control characters (octal 000-037) are printed as ^ <i>n</i> , where <i>n</i> is the corresponding ASCII character in the octal range 100-137 (@, A, B, C,..., X, Y, Z, [, \,], ^, and _); the DEL character (octal 0177) is printed as ^?. Other non-printable characters are printed as M- <i>x</i> , where <i>x</i> is the ASCII character specified by the low-order seven bits.
	When used with the -v option, the following options may be used:
-e	A \$ character will be printed at the end of each line prior to a new line.
-t	Tabs will be printed as ^I and form feeds will be printed as ^L
	The -e and -t options are ignored if the -v option is not specified.
-	Allows standard input to the cat command.

Exit Status

This command returns the following exit values:

0	All input files were output successfully.
>0	An error occurred.

Examples

Attention: Do not redirect output to one of the input files using the redirection symbol, > (caret).

1. To display a file at the workstation, enter:

```
cat notes
```

This command displays the data in the notes file.

2. To concatenate several files, enter:

```
cat section1.1 section1.2 section1.3 >section1
```

This command creates a file named `section1` that is a copy of `section1.1` followed by `section1.2` and `section1.3`.

3. To suppress error messages about files that do not exist, enter:

```
cat -q section2.1 section2.2 section2.3 >section2
```

If `section2.1` does not exist, this command concatenates `section2.2` and `section2.3`. The result is the same if you do not use the `-q` flag, except that the `cat` command displays the error message:

```
cat: cannot open section2.1
```

You may want to suppress this message with the `-q` flag when you use the `cat` command in shell procedures.

4. To append one file to the end of another, enter:

```
cat section1.4 >> section1
```

The `>>` (two carets) appends a copy of `section1.4` to the end of `section1`. If you want to replace the file, use the `>` (caret).

5. To add text to the end of a file, enter:

```
cat >>notes  
Get milk on the way home  
Ctrl-D
```

This command adds `Get milk on the way home` to the end of the file called `notes`. The `cat` command does not prompt; it waits for you to enter text. Press the `Ctrl-D` key sequence to indicate you are finished.

6. To concatenate several files with text entered from the keyboard, enter:

```
cat section3.1 - section3.3 >section3
```

This command concatenates the file `section3.1` with text from the keyboard (indicated by the minus sign), and the file `section3.3`, then directs the output into the file called `section3`.

Files

`/usr/bin/cat` Contains the `cat` command.

Related Information

The `cp` command.

cattracerpt command

Purpose

Formats a report from the trace log.

Syntax

```
cattracerpt [ -hookid List | -lshid ] [ -outfile FileName ]
```

Description

The **cattracerpt** command reads the trace log, formats the trace entries, and writes a report to standard output.

Flags

-hookid <i>List</i>	Limits report to hook IDs specified with the <i>List</i> variable. The <i>List</i> parameter items must be separated by commas.
-lshid	Displays the list of hook IDs. The cattracerpt -listid command can be used with the starttrace -event command that includes IDs of trace events.
-outfile <i>File</i>	Writes the report to a file instead of to standard output.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

- To send a trace report to the **newfile** file, enter:
cattracerpt -outfile newfile
- To display a list of hook IDs, enter:
cattracerpt -lshid

Related Information

The starttrace command, and the stoptrace command.

cfgassist command

Purpose

Allows administrators to manage some of the initial setup requirements of the Virtual I/O Server.

Syntax

cfgassist

The **cfgassist** command is used by a padmin user to manage initial setup of the Virtual I/O Server. Initial setup includes the following tasks:

- Set date and time zone
- Change passwords
- Set system security
- Virtual I/O Server TCP/IP configuration

- Install and update software
- Storage management
- Devices
- Electronic Service Agent

The **cfgassist** command is menu driven and accessible only by the *padmin* user, for security reasons.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

To display network sockets, type:

```
cfgassist -h
```

Related Information

The **bootlist** command.

cfgdev command

Purpose

Configures devices in the Virtual I/O Server.

Syntax

```
cfgdev [ -dev Name ]
```

Description

The **cfgdev** command configures devices. If there are any devices detected that have no device software installed when configuring devices, the **cfgdev** command returns a warning message with the name or a list of possible names for the device package that must be installed. If the specific name of the device package is determined, it is displayed as the only package name, on a line below the warning message. If the specific name cannot be determined, a colon-separated list of possible package names is displayed on a single line. A package name or list of possible package names is displayed for each of the devices, if more than one device is detected without its device software.

The system displays the following warning message when devices without their device software are detected:

```
cfgdev: 0514-621 WARNING: The following device packages are
      required for device support but are not currently
      installed.
devices.pci.22100020
devices.pci.14101800
devices.pci.scsi:devices.pci.00100300:devices.pci.NCR.53C825
```

In this example, two devices were found that had missing software, and the **cfgdev** command displayed the names of the device packages that must be installed. A third device that also has missing software was found, but in this case, the **cfgdev** command displays several possible device package names.

Attention: To protect the Configuration database, the **cfgdev** command is not interruptible. Stopping this command before execution is complete could result in a corrupted database.

Flags

-dev *Name* Specifies the named device to configure along with its children.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

1. To configure detected devices attached to the **scsi0** adapter, type:

```
cfgdev -dev scsi0
```

Related Information

The **chdev** command, the **chpath** command, the **lsdev** command, the **lsmmap** command, the **lspath** command, the **mkpath** command, the **mkvdev** command, the **rmdev** command, and the **rmpath** command.

cfglnagg command

Purpose

Add or remove adapters from a Link Aggregation or change Link Aggregation attributes.

Syntax

```
cfglnagg { -add [ -backup ] | -rm } LinkAggregation Adapter
```

```
cfglnagg [ -f ] -attr Attribute=NewValue LinkAggregation ...
```

Description

This command adds adapters to a Link Aggregation or removes adapters from a Link Aggregation. This command can also be used to modify Link Aggregation attributes. It is not necessary to detach the Link Aggregation’s interface to add or remove adapters or modify most Link Aggregation attributes.

Changing the **use_jumbo_frame** attribute requires the Link Aggregation to be detached. If this attribute is specified, the **cfglnagg** command detaches the Link Aggregation before modifying the **use_jumbo_frame** attributes, then brings the Link Aggregation back up. The user is prompted to continue unless the **-f** flag is specified.

Use the **lsdev** command with the **-attr** flag to view all attributes associated with a Link Aggregation device, including adapters.

Note: To create a Link Aggregation see the **mkvdev** command.

Flags

-add	Adds the specified Adapter to the specified Link Aggregation. If the adapter must be added as a back up adapter, the -backup flag must be specified.
-attr	Specifies an attribute of the specified Link Aggregation.
-backup	Specifies that the <i>Adapter</i> is being added as a back up adapter.
-f	Instructs the cfglnagg command to not prompt the user.
-rm	Deletes the specified Adapter from the specified Link Aggregation. The specified adapter can be either a primary or back up adapter.

Parameters

<i>Adapter</i>	Specifies the adapter to add or delete.
<i>LinkAggregation</i>	Specifies the Link Aggregation.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

1. To add adapter **ent8** to Link Aggregation **ent3**, type:

```
cfglnagg -add ent3 ent8
```

Related Information

The **mktcpip** command, the **mkvdev** command, the **hostname** command, the **startnetsvc** command, the **stopnetsvc** command, the **netstat** command, the **entstat** command, the **cfgnamesrv** command, the **hostmap** command, the **traceroute** command, the **ping** command, the **optimizenet** command.

cfgnamesrv command

Purpose

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

Syntax

To Add a Name Server Entry

```
cfgnamesrv -add { -ipaddr IPAddress | -dname DomainName | -slist SearchList }
```

To Delete a Name Server Entry

```
cfgnamesrv -rm { -ipaddr IPAddress | -domain | -slist }
```

To Change a Name Server Entry

```
cfgnamesrv -ch DomainName
```

To Display a Name Server Entry

```
cfgnamesrv -ls [ -namesrv | -domain | -slist ]
```

To Create the Configuration Database File

```
cfgnamesrv -mk [ -ipaddr IPAddress [ -dname DomainName ] [ -slist SearchList ] ]
```

To Change a Search List Entry

```
cfgnamesrv -chslist SearchList
```

Description

The **cfgnamesrv** command adds or deletes domain name server entries for local resolver routines in the system configuration database. To add a name server entry, specify an Internet Protocol address and, optionally, a domain name.

The **cfgnamesrv** command can show one or all domain name server entries in the system configuration database. 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. One search entry can exist.

Flags

-add	Adds an entry to the system configuration database.
-ch <i>DomainName</i>	Changes the domain name in the system configuration database.
-chslst	Changes the search list.
-dname	Indicates that the command deals with the domain name entry.
-domain	Specifies that the operation is on the domain name. Use this flag with the -rm flag and the -ls flag.
-ipaddr <i>IPAddress</i>	Indicates that the command deals with a name server entry. Use dotted decimal format for the given IP address.
-ls	Shows all domain and name server entries in the configuration system database. If you use the -ipaddr flag, the cfgnamesrv command shows all name server entries. If you use the -domain flag, the cfgnamesrv command shows the domain name entry found in the database.
-mk	Creates the system configuration database.
-namesrv	Specifies that the -ls flag should print all name server entries.
-rm	Deletes an entry in the system configuration database. It must be used with the -ipaddr <i>IPAddress</i> flag or the -domain flag. The -ipaddr flag deletes a name server entry. The -domain flag deletes the domain name entry.
-slst	Specifies that the operation is on the search list. Use this flag with the -rm and -ls flag.
-slst <i>SearchList</i>	Changes the search list in the system configuration database.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

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

```
cfgnamesrv -add -dname abc.aus.century.com
```
2. To add a name server entry with IP address 192.9.201.1, type:

```
cfgnamesrv -add -ipaddr 192.9.201.1
```
3. To show all system configuration database entries related to domain name server information used by local resolver routines, type:

```
cfgnamesrv -ls
```

The output is given in the following format:

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

4. To set the search list **abc.aus.century.com xyz.aus.century.com**, type:
`cfgnamesrv -chslist abc.aus.century.com xyz.aus.century.com`

The `cfglnagg` command, the `entstat` command, the `lsnetsh` command, the `mktcpip` command, the `netstat` command, and the `optimizenet` command.

cfgsvc command

Purpose

Configures agent specified by the agent name.

Syntax

```
cfgsvc AgentName [-attr Attribute=Value... | -ls | -key]
```

Description

The `cfgsvc` command configures the agent specified, by an agent name. Use the `lssvc` command to obtain a list of agents. After configuring the agent, you can start and stop the agent using the `startsvc` and `stopsvc` commands. Each time you run the command, it overwrites previous configurations.

To find the values for required attributes for an agent, use the `-ls` flag. If you do not specify a required attribute when you specify this command, a message is displayed that indicates the missing required attribute.

Agents might create `ssh` keys as part of their configuration. You can display the `ssh` public key that is generated during configuration by using the `-key` flag.

Parameters

<code>-attr Attribute=Value</code>	Identifies an attribute and value that are associated with the configuration of an agent.
<code>-ls</code>	Lists all the attributes that can be changed for a particular agent configuration.
<code>-key</code>	Displays the <code>ssh</code> public key that is generated for a particular agent configuration.

Note: The `-key` flag is not valid for the IBM® TotalStorage® Productivity Center.

Agent names

The following agents can be managed by the Virtual I/O Server.

DIRECTOR_agent

Attribute names for the IBM Systems Director agent

Using the **DIRECTOR_agent** agent, you can view and track hardware configuration details of the system and monitor performance and use of critical components, such as processors, disks, and memory.

RESTART_ON_REBOOT

This is an optional configuration attribute for the IBM Systems Director agent. It designates whether the IBM Systems Director agent will be restarted if the Virtual I/O Server is rebooted.

Valid values are:

True
False

ITM_base

Attribute names for the IBM Tivoli® Monitoring (ITM) agents

ITM_premium

ITM_cec

The **ITM_base**, **ITM_premium**, and **ITM_cec** agents are IBM Tivoli Monitoring agents. These agents provide system information including I/O virtualization mappings and system utilization. The **ITM_cec** agent is only available to the Integrated Virtualization Manager. The agent name is case sensitive.

These agents require that you specify the **HOSTNAME** attribute for configuration. Also, a ssh public key is generated if one does not exist.

hostname

Provides the monitoring agent with a Tivoli Enterprise Management System (TEMS) host name. This is a required attribute.

Valid values are:

Host name
IP address

RESTART_ON_REBOOT

Designates whether the monitoring agent will be restarted if the Virtual I/O Server is rebooted. Because you can only run one monitoring agent on the Virtual I/O Server at any given time, configuring more than one monitoring agent to restart is only valid for the last agent that is configured to restart. Run the **Issvc** command using the ITM agent name to verify which agent restarts.

Valid values are:

True
False

managing_system

Provides the monitoring agent with a Hardware Management Console (HMC) host name. This attribute is not valid for the Integrated Virtualization Manager.

Valid values are:

[user_name@]Host name
IP address

TSM_base

Attribute names for the Tivoli Storage Manager agent

SERVERNAME

The host name of the Tivoli Storage Manager server to which the Tivoli Storage Manager client is associated.

SERVERIP

The IP address or domain name of the Tivoli Storage Manager server to which the Tivoli Storage Manager client is associated.

NODENAME

The name of the machine on which the Tivoli Storage Manager client is installed.

ITUAM_base

Attribute names for the Tivoli Usage and Accounting Manager agent

ACCT_DATA0

The size, in megabytes, of the first data file that holds daily accounting information.

ACCT_DATA1

The size, in megabytes, of the second data file that holds daily accounting information.

ISYSTEM

The time, in minutes, when the agent generates system interval records.

IPROCESS

The time, in minutes, when the system generates aggregate process records.

Attribute names for the TotalStorage Productivity Center agents

The **TPC** agent is a TotalStorage Productivity Center agent. Agent names are case sensitive. This agent requires that you specify the **S**, **A**, **devAuth**, and **caPass** attributes for configuration. By default, specifying this agent configures both the **TPC_data** and **TPC_fabric** agents.

S Provides the TotalStorage Productivity Center agent with a TotalStorage Productivity Center server host name. This a required configuration attribute.

The valid values are:

Host name

IP address

A Provides the TotalStorage Productivity Center agent with an agent manager host name. This is a required attribute.

The valid values are:

Host name

IP address

devAuth

Sets the TotalStorage Productivity Center device server authentication password. This is a required attribute.

The valid value is:

Password

caPass Sets the CA authentication password. This is a required attribute.

The valid value is:

Password

caPort Sets the CA port. This is an optional attribute. The default value is 9510.

The valid value is:

Number

amRegPort

Specifies the agent manager registration port. This is an optional attribute. The default value is 9511.

The valid value is:

Number

amPubPort

Specifies the agent manager public port. This is an optional attribute. The default value is 9513.

The valid value is:

Number

dataPort

Specifies the TotalStorage Productivity Center data server port. This is an optional attribute. The default value is 9549.

The valid value is:

Number

devPort	Specifies the TotalStorage Productivity Center device server port. This is an optional attribute. The default value is 9550. The valid value is: <i>Number</i>
newCA	This is an optional attribute. The default value is true. The valid values are: True False
oldCA	This is an optional attribute. The default value is true. The valid values are: True False
daScan	This is an optional attribute. The default value is true. The valid values are: True False
daScript	This is an optional attribute. The default value is true. The valid values are: True False
daInstall	This is an optional attribute. The default value is true. The valid values are: True False
faInstall	This is an optional attribute. The default value is true. The valid values are: True False
U	Specifies to uninstall the agent. This is an optional attribute. The valid values are: All data fabric

Examples

IBM Tivoli Monitoring

1. To list all the attributes associated with an agent configuration, type the following command:

```
cfgsvc -ls ITM_base
```

2. To configure the ITM_base agent with several attributes, type the following command:

```
cfgsvc ITM_base -attr Restart_On_Reboot=TRUE hostname=tems_server  
managing_system=hmc_console
```

Note: When the RESTART_ON_REBOOT attribute is set to TRUE, the ITM agent is restarted when the Virtual I/O Server is being rebooted. The hostname attribute specifies the Tivoli Enterprise Monitoring Server (TEMS) IP address or hostname set to *tems_server*. The managing_system attribute specifies the HMC IP address or hostname set to *hmc_console*.

3. To display the ITM_base agent ssh public key, type the following command:

```
cfgsvc -key ITM_base
```

IBM Tivoli Storage Manager

1. To list all the attributes associated with an agent configuration, type the following command:

```
cfgsvc -ls TSM_base
```

2. To configure the TSM_base agent with several attributes, type the following command:

```
cfgsvc TSM_base -attr SERVERNAME=tsm_server SERVERIP=1.127.0.1  
NODENAME=VIOS
```

IBM Tivoli Usage and Accounting Manager

1. To list all the attributes associated with an agent configuration, type the following command:

```
cfgsvc -ls ITUAM_base
```

2. To configure the ITUAM_base agent with several attributes, type the following command:

```
cfgsvc ITUAM_base -attr ACCT_DATA0=15 ACCT_DATA1=15  
ISYSTEM=60 IPROCESS=60
```

IBM TotalStorage Productivity Center

1. To list all TotalStorage Productivity Center configurable attributes, type the following command:

```
cfgsvc TPC -ls
```

2. To configure TotalStorage Productivity Center with mandatory attributes, type the following command:

```
cfgsvc TPC -attr S=tpc_server_hostname  
A=agent_manager_hostname devAuth=password caPass=password
```

IBM Systems Director Agent

1. To list all the attributes associated with an agent configuration, type the following command:

```
cfgsvc -ls DIRECTOR_agent
```

2. To configure the DIRECTOR_agent with several attributes, type the following command:

```
cfgsvc DIRECTOR_agent -attr RESTART_ON_REBOOT=TRUE
```

Related Information

The **lssvc** command, the **startsvc** command, and the **stopsvc** command.

For more information about the various agents, see the following information:

- IBM Tivoli software and the Virtual I/O Server
- Configuring the IBM Tivoli agents and clients on the Virtual I/O Server
- IBM Systems Director software
- Configuring the IBM Systems Director agent

chbdsp command

Purpose

Change the characteristics of a backing device within a storage pool.

Syntax

Increase the size of a backing device:

```
chbdsp [-sp StoragePool | -bd BackingDevice -size Size
```

Rename a backing device:

chbdsp [-sp *StoragePool* | -bd *BackingDevice* -mv *NewName*

Description

The **chbdsp** command changes attributes of the specified backing device. If the **-size** flag is specified the backing device's size is increased. Size can be given in megabytes (###M/m) or gigabytes (###G/g).

If the **-mv** flag is specified, the backing device is renamed.

Note: The backing device that you specified cannot be assigned to a shared memory pool (to be used as a paging space device by a shared memory partition).

Flags

-bd <i>BackingDevice</i>	Specifies the backing device.
-mv <i>NewName</i>	Specifies the new name of the backing device.
-size <i>Size</i>	Specifies the minimum amount of storage to add to the backing device.
-sp <i>StoragePool</i>	Specifies the storage pool to be used.

Exit Status

23	Specified storage pool is not a valid storage pool.
39	The given backing device is not in the specified storage pool.
40	File backing device names cannot exceed "38" characters in length.

Examples

1. To increase the size of a the backing device *client1*, in the default storage pool by 3 gigabytes, type:
`chbdsp -bd client1 -size 3g`
2. To rename the backing device *cl_data*, in the storage pool *partition_3*, to *bank_data*, type:
`chbdsp -sp partition_3 -bd cl_data -mv bank_data`

chdate command

Purpose

Displays or changes the date, time or time zone.

Syntax

`chdate [mdddHHMM [YYyy | yy]] [-timezone TZ]`

`chdate [-year YYyy] [-month mm] [-day dd] [-hour HH] [-minute MM] [-timezone TZ]`

Description

Displays or changes the system date, time, or time zone. Changes made to the time zone will not take effect until the user logs out. For the time zone to take effect for the entire system, the system must be rebooted. All flags are optional, and the current system time information will be used if a flag was unspecified.

If no flags or arguments are specified, the **chdate** command displays the current date and time.

The *mdddHHMM* [*YYyy* | *yy*] parameters correspond to month, day, hour, minute, and optional 4 or 2 digit year.

Note: The user must have **padmin** authority to change the date and time.

Flags

-year	Sets the year to <i>YYyy</i> .
-month	Sets the month to <i>mm</i> .
-day	Sets the day to <i>dd</i> .
-hour	Sets the hour to <i>HH</i> in 24-hour format.
-minute	Sets the minute to <i>MM</i> .
-timezone	Sets the time zone (for example, CST6CDT).

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

1. To display the current date and time, type:
chdate
2. To change the date to Tue Oct 12 16:30:00 CDT 2004 for a system in the US Central time zone, type:
chdate -day 12 -month 10 -year 2004 -hour 16 -minute 30

or
chdate 101216302004

Note: The user must have **padmin** authority to change the date and time.

chdev command

Purpose

Changes the characteristics of a device.

Syntax

```
chdev -dev Name [ -perm ] [ -attr Attribute=Value [ Attribute=Value... ] ]
```

Description

The **chdev** command changes the characteristics of the device specified with the given device logical name (the **-dev *Name*** flag). Some changes may not be allowed when the device is in the **Available** state.

When the **-perm** flag is not specified, the **chdev** command applies the changes to the device and updates the database to reflect the changes. If the **-perm** flag is specified, only the database is updated to reflect the changes, and the device itself is left unchanged. This is useful in cases where a device cannot be changed because it is in use; in which case, the changes can be made to the database with the **-perm** flag, and the changes will be applied to the device when the system is restarted. Not all devices support the **-perm** flag.

Attention: To protect the Configuration database, the **chdev** command is not interruptible. To stop this command before execution is complete could result in a corrupted database.

Flags

-attr <i>Attribute=Value</i>	Specifies the device attribute value pairs used for changing specific attribute values. The <i>Attribute=Value</i> parameter can use one attribute value pair or multiple attribute value pairs for one -attr flag. If you use an -attr flag with multiple attribute value pairs, the list of pairs must be separated by spaces. For example, entering -attr Attribute=Value lists one attribute value pair per flag, while entering -attr Attribute1=Value1 Attribute2=Value2 lists more than one attribute value pair.
-dev <i>Name</i>	Specifies the device logical name, indicated by the <i>Name</i> parameter, whose characteristics are to be changed.
-perm	Changes the device's characteristics permanently without actually changing the device. This is useful for devices that cannot be made unavailable and cannot be changed while in the available state. By restarting the system, the changes will be applied to the device. Not all devices support the -perm flag.

Exit Status

See "Exit status for Virtual I/O Server commands" on page 1.

Examples

1. To change the retention instructions of the 4mm SCSI tape drive **rmt0** so that the drive does not move the tape to the beginning, then to the end, and then back to the beginning each time a tape is inserted or the drive is powered on, type:

```
chdev -dev rmt0 -attr ret=no
```

2. To change the SCSI ID of the available SCSI adapter **scsi0** that cannot be made unavailable or changed due to available disk drives connected to it, type:

```
chdev -dev scsi0 -attr id=6 -perm
```

To apply the change to the adapter, shutdown and restart the system.

Related Information

The `cfgdev` command, the `chpath` command, the `lsdev` command, the `lsmmap` command, the `lspath` command, the `mkpath` command, the `mkvdev` command, the `rmdev` command, and the `rmpath` command.

IVM chhwres command

Purpose

Changes hardware resources.

Syntax

To add, remove, or modify physical or virtual I/O, memory, or processors settings

```
chhwres -r io | mem | proc | virtualio | mempool [--subtype scsi | slot | eth | fc | pgdev] [-m <managed system>] -o a | r | s | rs [-p <partition name>] | --id <partition ID>] [-l <DRC index>] [-s <virtual slot number>] [-q <quantity>] [--procs <quantity>] [--procunits <quantity>] [-w <wait time>] [-d <detail level>] [--force] [-a "<attributes>"] [--entitled <size>] [--device] [--help]
```

To add, remove, or modify Host Ethernet Adapter resource assignments and settings

```
chhwres -r hea [--subtype slot | eth] [-m <managed system>] -o a | r | s [-p <partition name>] | --id <partition ID>] [-l <HEA DRC index>] [-g <port group ID>] [--physport <physical port ID>] [--logport]
```

<logical port ID>] [-w <wait time>] [-d <detail level>] [--force] [-a"<attributes>"] [--help]

Description

The **chhwres** command changes the hardware resource configuration of the managed system. The **chhwres** command is used to perform dynamic LPAR operations.

Flags

--device	Specifies the paging space device to add or remove. The --device flag is valid only with -r mempool --rsubtype pgdev .
-r	Specifies the type of hardware resources to change. Valid values are io for physical I/O, hea for Host Ethernet Adapter resources, virtualio for virtual I/O, mem for memory, and proc for processing resources.
--subtype	Specifies the subtype of hardware resources to change. Valid values for --rsubtype depend on the resource type you specify. <ul style="list-style-type: none">• io: slot, taggedio• virtualio: eth, scsi, fc, hsl, virtualopti• mempool: pgdev
-m <i>managed system</i>	Specifies the name of the managed system for which the hardware resource configuration is to be changed. <p>The name may either be the user-defined name for the managed system, or be in the form <i>ttt-mmm*sssssss</i>, where <i>ttt</i> is the machine type, <i>mmm</i> is the model, and <i>sssssss</i> is the serial number of the managed system.</p>
-o	Specifies the operation to perform. Valid values are a to add hardware resources to a logical partition, r to remove hardware resources from a logical partition, and s to set hardware resource related attributes for a logical partition an Host Ethernet Adapter, or the managed-system.
-p	Specifies the name of the logical partition for which the operation is to be performed. <p>You can either use this option to specify the name of the logical partition for which the operation is to be performed, or use the --id option to specify the logical partition's ID. The -p and the --id options are mutually exclusive.</p> <p>The logical partition name or --id is required for virtual Ethernet.</p>
--id <i>partition ID</i>	Specifies the ID of the logical partition for which the operation is to be performed. To perform an add or remove operation, the logical partition must be in the running state. <p>You can either use this option to specify the ID of the logical partition for which the operation is to be performed, or use the -p option to specify the logical partition's name. The --id and the -p options are mutually exclusive.</p> <p>A logical partition is required to be specified with this option or the -p option for all operations except a virtual Ethernet, memory, Host Ethernet Adapter set operation.</p>
-l	When adding, removing, or moving a physical I/O slot, use this option to specify the DRC index of the slot. <p>When performing an Host Ethernet Adapter operation, use this option to specify the adapter ID of the Host Ethernet Adapter for which the operation is to be performed.</p> <p>This option is not valid for any other operation.</p>

-s	Specifies the virtual slot number of the virtual I/O adapter to add or remove. When adding a virtual I/O adapter, if this option is not specified then the next available virtual slot number will be assigned to the virtual I/O adapter. When removing a virtual I/O adapter, this option is required.
-q	Specifies the quantity of memory to add or remove. The quantity specified must be in megabytes, it must be a multiple of the memory region size for the managed-system, and it must be greater than 0.
--procs <i>quantity</i>	<p>When adding or removing processing resources to or from a logical partition using dedicated processors, use this option to specify the quantity of dedicated processors to add or remove.</p> <p>When adding or removing processing resources to or from a logical partition using shared processors, use this option to specify the quantity of virtual processors to add or remove.</p> <p>The quantity of processing resources specified with this option must be a whole number greater than 0.</p>
--procunits	When adding or removing processing resources to or from a logical partition using shared processors, use this option to specify the quantity of processing units to add or remove.
--physport	<p>Otherwise, the quantity of processing units specified with this option can have up to 2 decimal places. In either case, the quantity specified must be greater than 0.</p> <p>Specifies the ID of the Host Ethernet Adapter physical port. This option is required when adding an Host Ethernet Adapter logical port to a logical partition. This option is also required when setting Host Ethernet Adapter physical port attributes. This option is not valid for any other operation.</p>
--logport	Specifies the ID of the Host Ethernet Adapter logical port to add or remove. This option is required for a Host Ethernet Adapter add or remove operation. This option is not valid for any other operation.
-g	Specifies the Host Ethernet Adapter port group. This option is required for all Host Ethernet Adapter operations, and is not valid for any other operation.
-w	<p>Specifies the elapsed time, in minutes, after which an add, remove, or move operation will be stopped. <i>wait-time</i> must be a whole number. If wait-time is 0, the operation will not be timed out.</p> <p>If this option is not specified, a default value of 5 minutes is used.</p> <p>This option is valid for all add, remove, and move operations for AIX®, Linux®, and Virtual I/O Server logical partitions. This option is also valid for memory add, remove, and move operations for IBM i logical partitions.</p>
-d	<p>Specifies the level of detail to be displayed upon return of an add or remove operation. Valid values are 0 (none) through 5 (highest).</p> <p>If this option is not specified, a default value of 0 is used.</p> <p>This option is valid for all add or remove operations for AIX, Linux, and Virtual I/O Server logical partitions.</p>
--entitled	Dynamically changes the amount of I/O entitled memory that is assigned to the logical partition. Use the -o parameter to specify which operation to perform.
--force	Forces an add or remove operation to be performed for an AIX, Linux, and Virtual I/O Server logical partition that does not have an RMC connection to the Integrated Virtualization Manager. If this command completes successfully, you will need to restart your operating system for the change to take effect. Use this option only if you intentionally configured your LAN to isolate the Integrated Virtualization Manager from the operating system of your logical partition.

-a attributes

Specifies the configuration data needed to create virtual I/O adapters or set hardware resource related attributes. The configuration data consists of attribute name/value pairs, which are in comma separated value (CSV) format. The configuration data must be enclosed in double quotation marks. Possible values are:

adapter_type

Whether the adapter is a client or server. Valid values are client and server. IVM requires the value to be client. This attribute is optional.

addl_vlan_ids

alt_console_slot

The location of the physical I/O slot that contains the alternate console device for the IBM i logical partition. The default value is none.

alt_restart_device_slot

The location of the virtual I/O slot that contains the alternate restart device for the IBM i logical partition. If the load source slot has a value other than none, then this attribute is optional. Valid values are:

- Slot number (virtual I/O)
- none

conn_speed

The configured connection speed of the port in Mbps. Possible values are:

- auto
- 10
- 100
- 1000
- 10000

console_slot

The location of the virtual I/O slot that contains the console device for the IBM i logical partition. Valid values are:

- Slot number (virtual I/O)
- none

flow_control

Receive flow control value of the port

- 1 (enabled)
- 0 (disabled)

hsl_pool_id

Whether the logical partition participates in the high-speed link (HSL) pool. The default value is 0 (does not participate).

ieee_virtual_eth

Valid values:

- 0 - not IEEE 802.1Q compatible
- 1 - IEEE 802.1Q compatible

is_trunk

Valid values:

- 0 - no
- 1 - yes

lhea_capabilities

Comma-separated list of logical Host Ethernet Adapter capabilities, with each capability having one of the following formats: *adapter-ID/capability* or *adapter-ID/5/ieq/nieq/qp/cq/mr* where *ieq* (interruptible event queues), *niesq* (non-interruptible event queues), *qp* (queue pairs), *cq* (completion queues), and *mr* (memory regions) each specify the resource amount in addition to the base minimum. Valid values are:

- 0 - minimum
- 1 - low
- 2 - medium
- 3 - high
- 4 - dedicated
- 5 - custom

load_source_slot

The location of the virtual I/O slot that contains the load source for the IBM i logical partition. If the *alt_restart_device_slot* has a value other than none, then this attribute is optional. Valid values are:

- Slot number (virtual I/O)
- none

max_pool_mem

Maximum amount of physical memory that can be assigned to the memory pool without taking the pool offline. The value should be designated in megabytes and must be a multiple of the logical memory block (LMB) size. This attribute is required. The attribute supports the *=/+=/=-* format.

max_rcv_packet_size

Maximum transmission unit (MTU) size of the port. Possible values:

- 1500 (default)
- 9000 (jumbo)

mem_weight

The shared memory weight of the shared memory partition. This flag is used for determining the priority of logical partitions in a memory pool for distribution of memory. This attribute is only supported when the value of the *-o* parameter is *s*. This attribute supports the *=* format.

op_console_slot

The location of the virtual I/O slot that contains the directly-attached operations console device for the IBM i logical partition. The default value is none.

paging_storage_pool

The name of the default paging storage pool. This is the storage pool from which new paging space devices are created. This attribute is optional, but can only be defined during the creation stage.

paging_vios_ids

A comma-separated list of logical partition IDs of paging service partitions to associate with this memory pool. This attribute is optional, but if specified, must be equal to 1.

paging_vios_names

A comma-separated list of logical partition names of paging service partitions to associate with this memory pool. This attribute is optional, but if specified, must be equal to the name of the Virtual I/O Server logical partition.

pend_port_group_mcs_value

A Host Ethernet Adapter port group attribute value. In order to synchronize the pending value with the current value, you need to reboot the system.

pool_mem

The amount of physical memory to assign, to add, or remove from the memory pool in megabytes. This required value must be a multiple of the logical memory block size.

The attribute supports the `=/+=/-=` format.

port_vlan_id

This option is required if you are adding a virtual Ethernet adapter.

promisc_lpar_id

The ID assigned to the promiscuous logical partition. Possible values:

- none
- 1 - 254

promisc_lpar_name

Name assigned to the promiscuous logical partition.

sharing_mode

The sharing mode of the logical partition.

Valid values are:

keep_idle_procs: Never share processors

share_idle_procs: Share processors only when logical partition is inactive

share_idle_procs_always: Always share processors

share_idle_procs_active: Share processors only when the logical partition is active

cap: Capped mode. **uncap:** Uncapped mode.

remote_lpar_id

The ID of the logical partition that has the virtual SCSI or virtual fibre channel server adapter.

A value of any indicates that any client adapter should be allowed to connect to this adapter.

Integrated Virtualization Manager (IVM) requires the value to be 1. This attribute is optional.

Note: The **remote_lpar_id** attribute and the **remote_lpar_name** attribute are mutually exclusive.

remote_lpar_name

Specifies a user-defined name for the logical partition where the virtual fibre channel or virtual SCSI server adapter will be created.

IVM will require the value of this to be the name of the Virtual I/O Server logical partition. This attribute is optional.

Note: The **remote_lpar_id** attribute and the **remote_lpar_name** attribute are mutually exclusive.

- remote_slot_num**
Specifies the slot number where the virtual fibre channel or virtual SCSI server adapter will be created on the remote logical partition.
If the value is not specified, IVM will use the next available slot. This attribute is optional.
- trunk_priority**
Valid values are integers between 1 and 15, inclusive. Required for a trunk adapter.
- uncap_weight**
The weighted average of processing priority when in uncapped sharing mode. The smaller the value, the lower the weight. Possible values are 0 - 255.
- virtual_opti_pool_id**
The location of the virtual I/O slot that contains the directly-attached operations console device for the IBM i logical partition. The default value is 0 (does not participate).
- vlan_id_list**
List of virtual LAN IDs that the logical port has access to.
- wwpns** Designates the worldwide port names to this adapter, which are entered as a comma-separated list. If the worldwide port names are not specified, IVM will request that a pair be allocated. Each worldwide port name must be a 16-character hexadecimal value, and two worldwide port names must be specified.
- help** Displays the help text for this command and exit.

Exit Status

This command has a return code of 0 on success.

Examples

- To change the tagged I/O for an IBM i logical partition, type the following command:

```
chhwres -r io -rsubtype taggedio -o s --id <LPAR ID> | -p <LPAR NAME>
-a ATTR=VALUE
```
- Configure a Host Ethernet Adapter port for a logical partition:
 - Add a Host Ethernet Adapter port to the logical partition with ID 2, with port 3 in port group 1:

```
chhwres -r hea -o a -l 23000000 -g 1 -a vlan_id_list=all
--id 2 --logport 3 --physport 0
```
 - Set the connection speed of physical port 0 to *auto*:

```
chhwres -r hea -o s -l 23000000 -g 1 -a conn_speed=auto --physport 0
```
- Add a virtual Ethernet adapter to the management partition in slot 25 with VLAN tags on VLAN 212 and VLAN 313 to be used with a shared Ethernet adapter:

```
chhwres -r virtualio --rsubtype eth -o a --id 1 -s 25
-a port_vlan_id=2,ieee_virtual_eth=1,
\"addl_vlan_ids=212,313\",is_trunk=1,trunk_priority=1
```
- Add a virtual Ethernet adapter to partition 4 in virtual slot 5 with VLAN tags on VLAN 212 and VLAN 313:

```
chhwres -r virtualio --rsubtype eth -o a --id 4 -s 5
-a port_vlan_id=2,ieee_virtual_eth=1,
\"addl_vlan_ids=212,313\"
```
- Create a memory pool with 4 GB of physical memory in the pool and 8 GB of maximum memory in the pool. Use the rootvg volume group to create paging space devices:

```
chhwres -r mempool -o a
-a pool_mem=4096,max_pool_mem=8192,paging_storage_pool=rootvg
```

chlang command

Purpose

Changes the language settings for the system.

Syntax

```
chlang { [ -msg msgtran ] -lang Name [ -dev Media | -ls }
```

Description

The **chlang** command is a high-level command that changes the language and keyboard settings for the entire Virtual I/O Server. The user needs to log out for language changes to take effect. If the language file sets are not already installed on the system, the **-dev** flag is used to specify their location.

When **chlang** is run with the **-ls** option, all available languages are listed.

Flags

-msg <i>msgtran</i>	Modifies the NSLPATH environment variable. The <i>msgtran</i> parameter is a colon-separated list of message translations (locale names) that indicates the message translation hierarchy required for the system or user.
-dev <i>Media</i>	Specifies the device or directory containing the images to install.
-lang <i>Name</i>	Specifies the language-territory (locale name) that will become the locale setting for the LANG environment variable.
-ls	Lists available languages.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

1. To change the language for the entire system to French Canadian, type:

```
chlang -lang fr_CA
```
2. To display available languages:

```
chlang -ls
```

Related Information

The license command.

IVM chled command

Purpose

Change states for physical and virtual LEDs. This command is available only in an Integrated Virtualization Manager environment.

Syntax

To change virtual partition LEDs:

```
chled -r sa -t virtuallpar -o Operation { -p PartitionName | --id PartitionID } [ -m ManagedSystem ]
```

To change virtual system LEDs:

```
chled -r sa -t virtualsys -o Operation [ -m ManagedSystem ]
```

Description

The **chled** command changes the states of physical and virtual LEDs.

Flags

-r <i>ResourceType</i>	The type of LED resource to change. The only valid value is sa for System Attention (SA) LED.
-t <i>SystemAttentionType</i>	The type of System Attention (SA) LEDs to change.
	-r sa -t virtuallpar
	Change attributes for virtual partition system attention LEDs Attributes: lpar_id, lpar_name, state Filters: None
	-r sa -t virtualsys
	Change attributes for virtual system attention LEDs Attributes: state Filters: None
-o	The operation to perform on the LED. Valid values are: off deactivate the LED on activate the LED
-m <i>ManagedSystem</i>	The name of the managed system. This attribute is optional because there is only one system to manage. The name may either be the user-defined name for the managed system, or be in the form ttt-mmm*sssssss, where ttt is the machine type, mmm is the model, and sssssss is the serial number of the managed system.
-p <i>PartitionName</i>	The name of the partition for which the virtual partition system attention LED is to be changed.
--id <i>PartitionID</i>	The ID of the partition for with the virtual partition system attention LED is to be changed.

Exit Status

This command has a return code of 0 on success.

Security

This command is not accessible by users with the ViewOnly role.

Examples

1. To deactivate the virtual system attention LED for the system, type:

```
chled -r sa -t virtualsys -o off
```

2. To activate the virtual partition system attention LED for partition lpar3, type:
`chled -r sa -t virtualpar -o on -p lpar3`
3. To deactivate the virtual partition system attention LED for the partition with ID 3, type:
`chled -r sa -t virtualpar -o off --id 3`

Related Information

The `Isled` command.

IVM `chlparutil` command

Purpose

Change settings for data collection.

Syntax

```
chlparutil -r config -s SampleRate [ -m ManagedSystem ]
```

Description

The `chlparutil` command changes the settings for data collection such as the sample rate at which data is collected.

Flags

<code>-r ResourceType</code>	The type of resources to change:
<code>-s SampleRate</code>	config Change configuration settings. The interval in seconds to sample utilization data. An interval of zero disables sampling. Valid sample rates are 0, 30, 60, 300, 1800, and 3600.
<code>-m ManagedSystem</code>	The name of the managed system. This attribute is optional because there is only one system to manage. The name may either be the user-defined name for the managed system, or be in the form <code>ttt-mmm*sssssss</code> , where <code>ttt</code> is the machine type, <code>mmm</code> is the model, and <code>sssssss</code> is the serial number of the managed system.

Exit Status

This command has a return code of 0 on success.

Security

This command is not accessible by users with the `ViewOnly` role.

Examples

1. Disable the collection of utilization data:
`chlparutil -r config -s 0`

Related Information

The `Islparutil` command.

chlv command

Purpose

Changes the characteristics of a logical volume.

Syntax

To Change the Name of a logical volume:

```
chlv -lv NewLogicalVolumeName LogicalVolume
```

Description

The **chlv** command changes the characteristics of a logical volume according to the command flags. The *LogicalVolume* parameter can be a logical volume name or logical volume ID. Changing the name of a logical volume is the only supported option.

Note: A logical volume that has been assigned as a backing device or as a paging space device cannot be renamed.

Flags

-lv	Specifies logical volume
-----	--------------------------

chmod command

Purpose

Changes file modes.

Syntax

To Change File Modes Symbolically

```
chmod [ -R ] [ -h ] [ -f ] [ [ u ] [ g ] [ o ] | [ a ] ] { - | + | = } [ r ] [ w ] [ x ] [ X ] [ s ] [ t ] { File ... | Directory ... }
```

To Change File Modes Numerically

```
chmod [ -R ] [ -h ] [ -f ] PermissionCode { File ... | Directory ... }
```

Description

The **chmod** command modifies the mode bits and the extended access control lists (ACLs) of the specified files or directories. The mode can be defined symbolically or numerically (absolute mode).

When a symbolic link is encountered and you have not specified the **-h** flag, the **chmod** command changes the mode of the file or directory pointed to by the link and not the mode of the link itself. If you specify the **-h** flag, the **chmod** command prevents this mode change.

If you specify both the **-h** flag and the **-R** flag, the **chmod** command descends the specified directories recursively, and when a symbolic link is encountered, the mode of the file or directory pointed to by the link is not changed.

Flags

- f** Suppresses all error reporting except invalid permissions and usage statements.
- h** Suppresses a mode change for the file or directory pointed to by the encountered symbolic link.
Note: This behavior is slightly different from the behavior of the **-h** flag on the **chgrp** and **chown** commands because mode bits cannot be set on symbolic links.
- R** Descends only directories recursively, as specified by the pattern *File...|Directory....*. The **-R** flag changes the file mode bits of each directory and of all files matching the specified pattern. See Example 6.

When a symbolic link is encountered and the link points to a directory, the file mode bits of that directory are changed but the directory is not further traversed.

Symbolic Mode

To specify a mode in symbolic form, you must specify three sets of flags.

Note: Do not separate flags with spaces.

The first set of flags specifies who is granted or denied the specified permissions, as follows:

- u** File owner.
- g** Group and extended ACL entries pertaining to the file's group.
- o** All others.
- a** User, group, and all others. The **a** flag has the same effect as specifying the **ugo** flags together. If none of these flags are specified, the default is the **a** flag and the file creation mask (umask) is applied.

The second set of flags specifies whether the permissions are to be removed, applied, or set:

- Removes specified permissions.
- +** Applies specified permissions.
- =** Clears the selected permission field and sets it to the permission specified. If you do not specify a permission following **=**, the **chmod** command removes all permissions from the selected field.

The third set of flags specifies the permissions that are to be removed, applied, or set:

- r** Read permission.
- w** Write permission.
- x** Execute permission for files; search permission for directories.
- X** Execute permission for files if the current (unmodified) mode bits have at least one of the user, group, or other execute bits set. The **X** flag is ignored if the *File* parameter is specified and none of the execute bits are set in the current mode bits.

Search permission for directories.
- s** Set-user-ID-on-execution permission if the **u** flag is specified or implied. Set-group-ID-on-execution permission if the **g** flag is specified or implied.
- t** For directories, indicates that only file owners can link or unlink files in the specified directory. For files, sets the **save-text** attribute.

Numeric or Absolute Mode

The **chmod** command also permits you to use octal notation for the mode. The numeric mode is the sum of one or more of the following values:

- 4000** Sets user ID on execution.
- 2000** Sets group ID on execution.

1000	Sets the link permission to directories or sets the save-text attribute for files.
0400	Permits read by owner.
0200	Permits write by owner.
0100	Permits execute or search by owner.
0040	Permits read by group.
0020	Permits write by group.
0010	Permits execute or search by group.
0004	Permits read by others.
0002	Permits write by others.
0001	Permits execute or search by others.

Note:

1. Specifying the mode numerically disables any extended Access Control Lists (ACL).
2. Changing group access permissions symbolically also affects the extended ACL entries. The group entries in the ACL that are equal to the owning group of the file are denied any permission that is removed from the mode.
3. You can specify multiple symbolic modes separated with commas. Operations are performed in the order they appear from left to right.
4. You must specify the mode symbolically or use an explicit 4-character octal with a leading zero (for example, 0755) when removing the set-group-ID-on-execution permission from directories.

Security

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

Only the owner of the file or the root user can change the mode of a file.

Exit Status

This command returns the following exit values:

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

Examples

1. To add a type of permission to several files:

```
chmod g+w chap1 chap2
```

This adds write permission for group members to the files chap1 and chap2.

2. To make several permission changes at once:

```
chmod go-w+x mydir
```

This denies group members and others the permission to create or delete files in mydir (**go-w**) and allows group members and others to search mydir or use it in a path name (**go+x**). This is equivalent to the command sequence:

```
chmod g-w mydir
chmod o-w mydir
chmod g+x mydir
chmod o+x mydir
```

3. To permit only the owner to use a shell procedure as a command:

```
chmod u=rwx,go= cmd
```

This gives read, write, and execute permission to the user who owns the file (**u=rwx**). It also denies the group and others the permission to access `cmd` in any way (**go=**).

If you have permission to execute the `cmd` shell command file, then you can run it by entering:

```
cmd
```

Note: Depending on the **PATH** shell variable, you may need to specify the full path to the `cmd` file.

4. To use Set-ID Modes:

```
chmod ug+s cmd
```

When the `cmd` command is executed, the effective user and group IDs are set to those that own the `cmd` file. Only the effective IDs associated with the child process that runs the `cmd` command are changed. The effective IDs of the shell session remain unchanged.

This feature allows you to permit access to restricted files. Suppose that the `cmd` program has the Set-User-ID Mode enabled and is owned by a user called `dbms`. The user `dbms` is not actually a person, but might be associated with a database management system. The user `betty` does not have permission to access any of `dbms`'s data files. However, she does have permission to execute the `cmd` command. When she does so, her effective user ID is temporarily changed to `dbms`, so that the `cmd` program can access the data files owned by the user `dbms`.

This way the user `betty` can use the `cmd` command to access the data files, but she cannot accidentally damage them with the standard shell commands.

5. To use the absolute mode form of the **chmod** command:

```
chmod 644 text
```

This sets read and write permission for the owner, and it sets read-only mode for the group and others. This also removes all extended ACLs that might be associated with the file.

6. To recursively descend directories and change file and directory permissions given the tree structure:

```
./dir1/dir2/file1
./dir1/dir2/file2
./dir1/file1
```

enter this command sequence:

```
chmod -R 777 f*
```

which will change permissions on `./dir1/file1`.

But given the tree structure of:

```
./dir1/fdir2/file1
./dir1/fdir2/file2
./dir1/file3
```

the command sequence:

```
chmod -R 777 f*
```

will change permissions on:

```
./dir1/fdir2
./dir1/fdir2/file1
./dir1/fdir2/file2
./dir1/file3
```

File

`/usr/bin/chmod` Contains the **chmod** command .

chpath command

Purpose

Changes the operational status of paths to a MultiPath I/O (MPIO) capable device, or changes an attribute associated with a path to an MPIO capable device.

Syntax

```
chpath -dev Name -op OpStatus [ -pdev Parent ] [ -conn Connection ]
```

```
chpath -dev Name -pdev Parent [ -conn Connection ] [ -perm ] -attr Attribute=Value...
```

Description

The **chpath** command either changes the operational status of paths to the specified device (the **-dev** *Name* flag) or it changes one, or more, attributes associated with a specific path to the specified device. The required syntax is slightly different depending upon the change being made.

The first syntax shown above changes the operational status of one or more paths to a specific device. The set of paths to change is obtained by taking the set of paths which match the following criteria:

- The target device matches the specified device.
- The parent device matches the specified parent (**-pdev** *Parent*), if a parent is specified.
- The connection matches the specified connection (**-conn** *Connection*), if a connection is specified.
- The path status is **PATH_AVAILABLE**

The operational status of a path refers to the usage of the path as part of MPIO path selection. The value of **enable** indicates that the path is to be used while **disable** indicates that the path is not to be used. It should be noted that setting a path to **disable** impacts future I/O, not I/O already in progress. As such, a path can be disabled, but still have outstanding I/O until such time that all of the I/O that was already in progress completes. As such, if **-op disable** is specified for a path and I/O is outstanding on the path, this fact will be displayed.

Disabling a path affects path selection at the device driver level. The **path_status** of the path is not changed in the device configuration database. The **lspath** command must be used to see current operational status of a path.

The second syntax shown above changes one or more path specific attributes associated with a particular path to a particular device. Note that multiple attributes can be changed in a single invocation of the **chpath** command; but all of the attributes must be associated with a single path. In other words, you cannot change attributes across multiple paths in a single invocation of the **chpath** command. To change attributes across multiple paths, separate invocations of **chpath** are required; one for each of the paths that are to be changed.

Flags

-attr *Attribute=Value*

Identifies the attribute to change as well as the new value for the attribute. The *Attribute* is the name of a path specific attribute. The *Value* is the value which is to replace the current value for the *Attribute*. The *Attribute=Value* parameter can use one attribute value pair or multiple attribute value pairs for one **-attr** flag. If you use an **-attr** flag with multiple attribute value pairs, the list of pairs must be enclosed in quotation marks with spaces between the pairs. For example, entering **-attr *Attribute=Value*** lists one attribute value pair per flag, while entering **-attr '*Attribute1=Value1 Attribute2=Value2*'** lists more than one attribute value pair.

-dev <i>Name</i>	Specifies the logical device name of the target device for the path(s) affected by the change. This flag is required in all cases.
-pdev <i>Parent</i>	Indicates the logical device name of the parent device to use in qualifying the paths to be changed. This flag is required when changing attributes, but is optional when change operational status.
-perm	Changes the path's characteristics without actually changing the path. The change takes effect on the path the next time the path is unconfigured and then configured (possibly on the next boot).
-conn <i>Connection</i>	Indicates the connection information to use in qualifying the paths to be changed. This flag is optional when changing operational status. When changing attributes, it is optional if the device has only one path to the indicated parent. If there are multiple paths from the parent to the device, then this flag is required to identify the specific path being changed.
-op <i>OpStatus</i>	Indicates the operational status to which the indicated paths should be changed. The operational status of a path is maintained at the device driver level. It determines if the path will be considered when performing path selection. The allowable values for this flag are: <ul style="list-style-type: none"> enable Mark the operational status as enabled for MPIO path selection. A path with this status will be considered for use when performing path selection. Note that enabling a path is the only way to recover a path from a failed condition. disable Mark the operational status as disabled for MPIO path selection. A path with this status will not be considered for use when performing path selection. This flag is required when changing operational status. When used in conjunction with the -attr Attribute=Value flag, a usage error is generated.

Exit Status

See "Exit status for Virtual I/O Server commands" on page 1.

Examples

- To disable the paths between **scsi0** and the **hdisk1** disk device, enter:

```
chpath -dev hdisk1 -pdev scsi0 -op disable
```

The system displays a message similar to one of the following:

```
paths disabled
```

or

```
some paths disabled
```

The first message indicates that all **PATH_AVAILABLE** paths from **scsi0** to **hdisk1** have been successfully enabled. The second message indicates that only some of the **PATH_AVAILABLE** paths from **scsi0** to **hdisk1** have been successfully disabled.

Related Information

The `cfgdev` command, the `chdev` command, the `lsdev` command, the `lsmmap` command, the `lspath` command, the `mkpath` command, the `mkvdev` command, the `rmdev` command, and the `rmpath` command.

chrep command

Purpose

Change the characteristics of the Virtual Media Repository.

Syntax

`chrep -size Size`

Description

The **chrep** command increases the size of the Virtual Media Repository by at least the amount specified with the **-size** flag. The actual amount of storage added to the repository is dependent on the allocation size of the parent storage pool. Use the **lssp** command to determine the parent storage pools allocation size. The specified flag can not be a negative number.

Size	Minimum file storage pool size
###M/m	###MB
###G/g	###GB

Flags

-size Size Specifies the minimum amount of storage to add to the file storage pool.

Examples

To increase the size of the Virtual Media Repository by at least 64 megabyte, type the following command:

```
chrep -size 64m
```

chsp command

Purpose

Change the characteristics of a storage pool.

Syntax

Increase the size of a file storage pool:

```
chsp -add [-sp StoragePool] -size Size
```

Add physical volume to a logical volume storage pool:

```
chsp -add [ -f ] [-sp StoragePool] PhysicalVolume...
```

Remove a physical volume from a logical volume storage pool :

```
chsp -rm [ -f ] [-sp StoragePool] PhysicalVolume...
```

Set storage pool as the default:

```
chsp -default StoragePool
```

Description

The **chsp** command adds and removes physical volumes represented by the *PhysicalVolume* parameter from a logical volume storage pool. Before adding physical volume, the **chsp** command checks to verify that they are not already in another volume group, storage pool, or assigned to a shared memory pool (to

be used as a paging space device by a shared memory partition). If the system detects a description area from a volume group or storage pool, the command will fail. If the **-f** flag is specified, the physical volume is added even if it contains a description area, unless it is a member of another storage pool, volume group, or is assigned to a shared memory pool (to be used as a paging space device by a shared memory partition).

Before removing physical volumes from a logical volume the storage pool, the **chsp** command prompts the user to verify each logical volume should be removed. If the user specifies the **-f** flag, the logical volumes are removed with out prompting for input. If all physical volumes in a storage pool are removed, the storage pool is also removed.

Note: A logical volume storage pool which contains file storage pools or the Virtual Media Repository cannot be removed.

If the **-sp** flag is not specified, the default storage pool is assumed.

If the specified, or default, storage pool is a file storage pool and the **-size** flag is specified the given pools size will be increased by at least the amount specified. The specified flag cannot be a negative number and file storage pools cannot be reduced in size.

Size	Minimum file storage pool size
###M/m	###MB
###G/g	###GB

If the **-default** flag is included, the storage pool specified will become the default storage pool for all users. If no default has been set, the rootvg storage pool will be the default. Only the padmin user can set the default storage pool.

Files

-add	Adds the given physical volume into specified storage pool. If a storage pool is not specified the physical volume is added to the default pool.
-default	Sets the specified storage pool as the system default pool. Only the padmin user can set the default storage pool.
-f	When combined with the -add flag, forces the physical volume to be added to the specified storage pool unless the physical volume is part of another storage pool or volume group in the Device Configuration Database or a volume group that is active. When combined with the -rm flag, forces the removal of all logical volumes on the physical volume before removal from the storage pool.
-rm	Remove the specified physical volume from specified storage pool. If a storage pool is not specified, the physical volume is removed from the default pool. The user will be prompted to confirm the removal of any logical volumes on the physical volume.
-size <i>StoragePool</i>	Specifies the minimum amount of storage to add to the file storage pool.
-sp <i>StoragePool</i>	Specifies the storage pool to be changed.

Exit Status

23	Specified storage pool is not valid
----	-------------------------------------

Examples

1. To add physical volume hdisk3 to the default storage pool, type:
chsp -add hdisk3
2. To remove physical volume hdisk2 from clstorage storage pool, type:

```
chsp -rm -sp clstorage hdisk2
```

3. To increase the size of the file storage pool clientData by at least 1 gigabyte, type:

```
chsp -add -sp clientData -size 1g
```

IVM chsvcevent command

Purpose

Changes an existing serviceable event. This command is available only in an Integrated Virtualization Manager environment.

Syntax

To close an existing serviceable event:

```
chsvcevent -o close -p ProblemNumber -n Name -c CommentText [ -m ManagedSystem ]
```

Description

The **chsvcevent** command closes an existing serviceable event.

Flags

-o <i>Operation</i>	The operation to be performed. The only valid value is <i>close</i> .
-p <i>ProblemNumber</i>	The problem number (<i>problem_number</i>) for the event, as displayed by the lssvcevents command.
-n	A free form name string identifying the person who is closing the event.
-c	A free form text comment about why the event is being changed.
-m <i>ManagedSystem</i>	The name of the managed system. This attribute is optional because there is only one system to manage. The name may either be the user-defined name for the managed system, or be in the form <i>ttt-mmm*sssssss</i> , where <i>ttt</i> is the machine type, <i>mmm</i> is the model, and <i>sssssss</i> is the serial number of the managed system.

Exit Status

This command has a return code of 0 on success.

Security

This command is not accessible by users with the **ViewOnly** role.

Examples

1. To close a serviceable event, type:

```
chsvcevent -o close -p 6013EFFF-205F3F22-4CC992E5-F8B6270-7540D8A3  
-m 9111-520*XXXXXXX -n My Name -c Closing Comment
```

Related Information

The **lssvcevents** command, and the **mksvcevent** command.

IVM chsyscfg command

Purpose

Changes attributes of logical partitions, logical partition profiles, or the managed system. This command is available only in an Integrated Virtualization Manager environment.

Syntax

To change system attributes:

```
chsyscfg -r sys { -f ConfigurationFile | -i ConfigurationData } [ -m ManagedSystem ]
```

To change partition attributes

```
chsyscfg -r lpar { -f ConfigurationFile | -i ConfigurationData } [ -m ManagedSystem ]
```

To change partition profile attributes, the Host Ethernet Adapter logical port assignments, or the logical Host Ethernet Adapter capabilities

```
chsyscfg -r prof { -f ConfigurationFile | -i ConfigurationData } [ -m ManagedSystem ]
```

Description

The **chsyscfg** command changes attributes of logical partitions, logical partition profiles, or the managed system.

Flags

-r <i>ResourceType</i>	The type of resources to change: sys Managed system resources lpar Logical partition resources prof Logical partition profile resources
-m <i>ManagedSystem</i>	The name of the managed system. This attribute is optional because there is only one system to manage. The name may either be the user-defined name for the managed system, or be in the form tttt-mmm*sssssss, where tttt is the machine type, mmm is the model, and sssssss is the serial number of the managed system.

-f ConfigurationFile

The name of the file containing the configuration data needed to change the resources. The configuration data consists of attribute name and value pairs, which are in comma separated value (CSV) format. These attribute name and value pairs form a configuration record. A line feed marks the end of a configuration record. The file must contain one configuration record for each resource to be changed, and each configuration record must be for the same resource type. If the resource type is the managed system, then the file must contain only one configuration record.

The format of a configuration record is as follows:

```
attribute-name=value,attribute-name=value,...<LF>
```

Note that certain attributes accept a comma separated list of values, as follows:

```
"attribute-name=value,value,...",...<LF>
```

When a list of values is specified, the attribute name/value pair must be enclosed in double quotation marks. Depending on the shell being used, nested double quotation marks may need to be preceded by an escape character, which is usually a \ (back slash) character.

If '+=' is used in the attribute name/value pair instead of '=', then the specified value is added to the existing value for the attribute if the attribute is numerical. If the attribute is a list, then the specified values are added to the existing list.

If '-=' is used in the attribute name/value pair instead of '=', then the specified value is subtracted from the existing value for the attribute if the attribute is numerical. If the attribute is a list, then the specified values are deleted from the existing list.

Attribute names for logical partitions

allow_perf_collection

Valid values are:

- 0 - do not allow authority
- 1 - allow authority

ipl_source

The IPL source for the IBM i logical partition. This attribute is optional. Valid values are:

- a
- b
- c
- d

name | lpar_id

Name or ID of the logical partition to change (required)

new_name

New name for the logical partition.

work_group_id

Valid values are:

- none** - do not participate in the workload management group
- 1 - participate in the workload management group

Attribute names for logical partition profiles

alt_restart_device_slot

The location of the virtual I/O slot that contains the alternate restart device for the IBM i logical partition. If the load source slot is has a value other than none, then this attribute is optional. Valid values are:

- Slot Number (virtual I/O)
- none

auto_start

Valid values are:

- 0 - do not automatically start with system power on
- 1 - automatically start with system power on

boot_mode

Logical partition power on mode. Valid values are:

- norm** - normal
- dd** - diagnostic with default boot list
- ds** - diagnostic with stored boot list
- of** - Open Firmware OK prompt
- sms** - System Management Services

console_slot

The location of the virtual I/O slot that contains the console device for the IBM i logical partition. Valid values are:

- Slot Number (for virtual I/O)
- none

desired_io_entitled_mem

The amount of I/O entitled memory for a shared memory partition. This is the portion of memory that is reserved for I/O mappings. Valid values follow:

- auto (automatically manage)
- *Number of megabytes*

If the value is auto, the entitlement is calculated based on the virtual I/O configuration of the logical partition. If the virtual I/O configuration is changed, the entitlement is updated automatically. If auto is not used, no automatic adjustments are made. The default value is auto.

desired_mem

Assigned memory in megabytes

desired_procs

Assigned processors. In shared processing mode, this refers to virtual processors.

desired_proc_units

Assigned shared processing units.

lhea_capabilities

Comma-separated list of logical Host Ethernet Adapter capabilities, with each capability having one of the following formats: *adapter-ID/capability* or *adapter-ID/5/ieq/nieq/qp/cq/mr* where *ieq* (interruptible event queues), *niesq* (non-interruptible event queues), *qp* (queue pairs), *cq* (completion queues), and *mr* (memory regions) each specify the resource amount in addition to the base minimum. Valid values are:

- 0 - minimum
- 1 - low
- 2 - medium
- 3 - high
- 4 - dedicated
- 5 - custom

lhea_logical_ports

Comma separated list of Logical Host Ethernet Adapter (LHEA) logical ports, with each logical port having the following format:

*adapter-ID/port-group/physical-port-ID/
logical-port-ID/allowed-VLAN-IDs*

All 4 '/' characters must be present, but optional values may be omitted. Optional values are allowed-VLAN-IDs.

load_source_slot

The location of the virtual I/O slot that contains the load source for the IBM i logical partition. If the *alt_restart_device_slot* has a value other than none, then this attribute is optional. Valid values are:

- Slot number (virtual I/O)
- none

lpar_avail_priority

The priority of the logical partition to maintain its entitled processors. If a processor failure occurs, processing resources will be removed first from the lowest priority logical partition. Valid values are 0 - 255.

Note: The Virtual I/O Server logical partition must have a higher priority than any other logical partition on the system.

lpar_proc_compat_mode

The requested compatibility mode. Use **lssyscfg -r sys -F lpar_proc_compat_modes** to retrieve a list of valid values.

max_mem

Maximum memory in megabytes.

min_mem

Minimum memory in megabytes.

min_procs

Minimum processors. In shared processing mode, this refers to virtual processors.

max_procs

Maximum processors. In shared processing mode, this refers to virtual processors.

max_proc_units

Maximum shared processing units.

min_proc_units

Minimum shared processing units.

max_virtual_slots

Maximum number of virtual I/O adapter slots

mem_mode

Logical partition memory mode. Valid values follow:

- **ded**: dedicated processor mode
- **shared**: shared processor mode

If the memory mode is shared, the logical partition cannot be assigned any physical I/O slots or host Ethernet adapter resources, the `proc_mode` attribute on the logical partition must be shared, and a memory pool must exist.

Note: The Virtual I/O Server logical partition only supports the dedicated memory mode.

mem_weight

The shared memory weight of the shared memory partition. The memory weight is used to determine the priority of logical partitions in a memory pool for distributing memory. Valid values are 0 - 255. The default value is 128.

name | lpar_name | lpar_id

Name or ID of the logical partition to change (required)

Note: This command uses the profile name and `lpar_name` interchangeably because this environment does not support multiple profiles per logical partition. When using this command on the Hardware Management Console, you must specify the profile name and logical partition name or ID because it supports multiple profiles per logical partition.

new_name

New name for the logical partition.

paging_device

The paging space device to use if a memory pool is used. A paging space device is a block storage device that has been added to the memory pool and is not designated as a paging device for any other logical partition. This attribute is optional. If omitted, an appropriate paging device is selected automatically. If the `paging_device` value is a blank string and a paging space device is currently assigned to this partition, the paging space device is removed from the logical partition.

proc_mode

Valid values follow:

- **ded**: dedicated processor mode
- **shared**: shared processor mode

sharing_mode

The sharing mode of the logical partition. Valid values follow:

- **keep_idle_procs**: Never share processors
- **share_idle_procs**: Share processors only when the logical partition is inactive
- **share_idle_procs_always**: Always share processors
- **share_idle_procs_active**: Share processors only when the logical partition is active
- **cap**: Capped mode
- **uncap**: Uncapped mode

uncap_weight

A weighted average of processing priority when in uncapped sharing mode. The smaller the value, the lower the weight. Valid values are: 0 - 255

virtual_eth_adapters

Comma-separated list of virtual Ethernet adapters, with each adapter having the following format: *slot_number/is_ieee/port_vlan_id/additional_vlan_ids/is_trunk/is_required* All 5 '/' characters must be present, but optional values may be omitted. Optional values are `is_ieee`, `additional_vlan_ids`, `is_required` and `is_trunk`.

Valid values for `is_ieee`, `is_trunk`, and `is_required`

- 0 - no
- 1 - yes

For example, `4/0/2//0/0` specifies a virtual Ethernet adapter with a virtual slot number of 4, is not IEEE 802.1Q enabled, has a port virtual LAN ID of 2, no additional virtual LAN IDs, it is not a trunk adapter, and is not required.

virtual_fc_adapters

Comma-separated list of virtual fibre channel adapters. Each item in this list has the following format:

```
virtual_slot_num/adapter_type/remote_lpar_id/  
remote_lpar_name/remote_slot_num/wwpn_list/is_required
```

Required values: remote_lpar_id, remote_lpar_name, adapter_type, virtual_slot_num

Note: You can specify either remote_lpar_id, remote_lpar_name, or use them both, but at least one of the values is required.

Valid values for adapter_type:

- client
- server

Note: If you specify a value for adapter type, the Integrated Virtualization Manager (IVM) requires the adapter type to be a client.

Optional values: wwpn_list, is_required, remote_slot_num

When you add a virtual fibre channel adapter, wwpn_list can be left blank to allow IVM to automatically assign worldwide port names to the client adapter. If you leave wwpn_list blank and the virtual slot number specified for this adapter already contains a virtual fibre channel adapter, IVM will use the worldwide port names that are already assigned. New worldwide port names will only be generated if it is a new adapter. If you specify a wwpn_list value, there must be exactly two values. Each worldwide port name must be a 16-character hexadecimal value. These values are not case sensitive.

Valid values for is_required:

- 0 - no
- 1 - yes

A value of none or an empty string indicates that no virtual fibre channel adapters should be assigned.

Note: If the logical partition being changed is the Virtual I/O Server, you cannot change the current configuration. The IVM handles the client and server adapter as a pair; thus, IVM handles modifications automatically.

virtual_scsi_adapters

Comma-separated list of virtual SCSI adapters. Each item in this list has the format:

```
slot_num/adapter_type/remote_lpar_id/remote_lpar_name/  
remote_slot_num/is_required
```

The attribute names are not present in the list, just their values are present. If an attribute is optional and is not to be included, then no value is specified for that attribute. For example, `2/client//lpar2/3/0` specifies a virtual client SCSI adapter with a virtual slot number of 2, a server partition name of `lpar2`, a server slot number of 3, and is not required. The server partition ID was omitted.

Required values: `slot_num`, `adapter_type`, `remote_lpar_id`, `remote_lpar_name`

Note: You can specify `remote_lpar_id`, `remote_lpar_name`, or both, but at least one of the values is required.

Optional values: `is_required`, `remote_slot_num`

Note: IVM requires that virtual slot number 2 always contain a virtual SCSI adapter, so if you specify adapters in any other slots, the default adapter is still created in slot number 2. If you use the `chsyscfg` command with an empty list for the `virtual_scsi_adapters` attribute, all virtual SCSI adapters are removed, except the default adapter.

Valid values for `adapter_type`:

- `client`: client adapter
- `server`: server adapter, valid for Virtual I/O Server logical partitions only

Valid values for `is_required`:

- 0 - no
- 1 - yes

Attribute names for the managed system

lpar_comm_default

Returns the `lpar_comm_ipaddr` to using the default IP address configured on the system as reported by `lscpip` -interfaces. Valid values:

- 1: Default IP address will be used.

lpar_comm_ipaddr

The IP address through which client logical partitions will communicate with the management logical partition. This is used primarily for dynamic LPAR. It is defaulted to the first IP address available on your system, but can be manually set if desired. If you set this manually and then change your systems IP address, you must update this value.

Note: This attribute supports multiple IP addresses using a comma-separated list.

new_name

New name for the managed system

pend_configured_max_lpars

The maximum number of logical partitions which can be created after restarting the managed system.

This option is deprecated. Instead use:

```
chsyscfg -r prof
```

with attribute "`max_virtual_slots`" value for partition 1.

-i ConfigurationData This option allows you to enter configuration data on the command line, instead of using a file. Data entered on the command line must follow the same format as data in a file, and must be enclosed in double quotation marks.

When this option is used, only a single resource can be changed.

The **-i** and the **-f** options are mutually exclusive.

Exit Status

This command has a return code of 0 on success.

Security

This command is not accessible by users with the ViewOnly role.

Examples

1. To change the user defined name for the managed system, type:
`chsyscfg -r sys -i "new_name=sys1"`
2. To change logical partitions using the configuration data in the file `/tmp/lparfile`, type:
`chsyscfg -r lpar -f /tmp/lparfile`
3. To reduce a partition profile's assigned and minimum memory by 256 MB, type:
`chsyscfg -r prof -i "lpar_name=partition3,min_mem-=256,desired_mem-=256"`

Related Information

The `Issyscfg` command, the `mksyscfg` command, and the `rmsyscfg` command.

IVM chsysstate command

Purpose

Changes the state of a partition. This command is operable only in the Integrated Virtualization Manager environment.

Syntax

To activate a partition:

```
chsysstate -r lpar -o on { -n Name | --id PartitionID } [ -k KeylockPosition ] [ -b BootMode ] [ -m ManagedSystem ]
```

To perform a partition shut down using the **shutdown** command on the client operating system:

```
chsysstate -r lpar -o osshutdown { -n Name | --id PartitionID } [ -m ManagedSystem ]
```

To perform a delayed partition shut down (white button shut down):

```
chsysstate -r lpar -o shutdown { -n Name | --id PartitionID } [ -m ManagedSystem ]
```

To perform an immediate partition shutdown (operator panel function 8):

```
chsysstate -r lpar -o shutdown --immed { -n Name | --id PartitionID } [ -m ManagedSystem ]
```


To perform an immediate restart of a partition (operator panel function 3):

```
chsysstate -r lpar -o shutdown --immed --restart { -n Name | --id PartitionID } [ -m ManagedSystem ]
```

To perform a restart of a partition after initiating a dump (operator panel function 22):

```
chsysstate -r lpar -o dumprestart { -n Name | --id PartitionID } [ -m ManagedSystem ]
```

To change the keylock position for a partition:

```
chsysstate -r lpar -o chkey -k KeylockPosition { -n Name | --id PartitionID } [ -m ManagedSystem ]
```

Description

The **chsysstate** command changes the state of a partition. To avoid data loss, use the shutdown facilities provided by the operating system in the partition.

Flags

-r <i>ResourceType</i>	The type of resources to change: lpar Logical partition resources
-m <i>ManagedSystem</i>	The name of the managed system. This attribute is optional because there is only one system to manage. The name may either be the user-defined name for the managed system, or be in the form tttt-mmm*sssssss, where tttt is the machine type, mmm is the model, and sssssss is the serial number of the managed system.
-o <i>Operation</i>	The type of operation to perform: <ul style="list-style-type: none">• chkey: Changes the keylock position• consoleservice: Disables a remote service session for the IBM i partition (operator panel function 65) followed by the activation of the dedicated service tools for the IBM i partition (operator panel function 21).• dston: Activates dedicated service tools for the IBM i partition (operator panel function 21).• dumprestart: Restarts after initiating a dump• iopdump: Allows use of the IOP control storage dump (operator panel function 70). This operation is valid for IBM i partitions only.• iopreset: Resets or reloads the failed IOP (operator panel function 67). This operation is valid for IBM i partitions only.• on: Power on• osshutdown: Shuts down using the client partition's shutdown command. This is the safest shutdown option, so should be used if available (RMC state must be active).• remotedston: Enables a remote service session for the IBM i partition (operator panel function 66).• retrydump: Retries the dump on the IBM i partition and restarts the partition when the dump is complete (operator panel function 34).• shutdown: Shuts down• remotedstoffs: Disables a remote service session for the IBM i partition (operator panel function 65).
-i <i>IPL Source</i>	The IPL source that will be used when activating the IBM i partition. You can also use the "IVM chsyscfg command" on page 64 command to complete this task. If this flag is not specified, then the current IPL source will be used. Valid values are: <ul style="list-style-type: none">• a• b• c• d

-b <i>BootMode</i>	Override the current power on mode setting. Valid values are: <ul style="list-style-type: none"> • norm: normal • dd: diagnostic with default boot list • ds: diagnostic with stored boot list • of: Open Firmware OK prompt • sms: System Management Services
-k <i>KeylockPosition</i>	Keylock position. Valid values are: <ul style="list-style-type: none"> • norm: normal keylock • manual: manual keylock
--immed	Force the state change immediately.
--restart	Restart the partition. This flag is valid only if the --immed flag is also specified.

Exit Status

This command has a return code of 0 on success.

Security

This command is not accessible by users with the ViewOnly role.

Examples

1. To power on the partition with an ID of 2 and set the boot mode to System Management Services, type the following command:

```
chsysstate -r lpar -o on --id 2 -b sms
```
2. To shut down the partition with an ID of 3, type the following command:

```
chsysstate -r lpar -o shutdown --id 3
```
3. To immediately restart the partition with an ID of 3 using the client operating system's shutdown command, type the following command:

```
chsysstate -r lpar -o osshutdown --restart --immed --id 3
```
4. To select the IPL source before activating an IBM i logical partition, type the following command:

```
chsysstate -r lpar -o [-i <IPL Source>]
```

Related Information

The **lssyscfg** command, the **mksyscfg** command, the **rmsyscfg** command, and the **mkvt** command.

chtcip command

Purpose

Changes the Virtual I/O Server TCP/IP settings and parameters.

Syntax

To change a network interface:

```
chtcip [-interface Interface -inetaddr Address -netmask SubnetMask]
```

```
chtcip [-interface Interface -gateway -add New_gateway_address -remove Old_gateway_address]
```

```
chtcip -ip6-interface Interface [-inetaddr Address] [-plen pfixlen ] [-gateway]
```

Description

The `chtcpip` command changes the TCP/IP settings and configuration on the Virtual I/O Server.

Flags

<code>-add</code> <i>New_Gateway_Address</i>	Specifies the new default gateway address to add.
<code>-inetaddr</code> <i>Address</i>	Changes the IP address of the host. Specify the address in dotted decimal notation.
<code>-interface</code> <i>Interface</i>	Specifies a particular network interface, for example <code>en0</code> .
<code>-ip6</code>	Specifies to use the IPv6 version of this command. The <code>-ip6</code> flag allows you to change an existing IPv6 address, state, prefix length, and default gateway.
<code>-gateway</code> <i>Gateway</i>	Changes the gateway address for a static route. Specify both the current address and new address in dotted decimal notation.
<code>-netmask</code> <i>SubnetMask</i>	Specifies the subnet mask of the gateway, which is used to route through the appropriate subnetwork.
<code>-plen</code> <i>prefixLen</i>	Specifies the prefix length of the IPv6 interface.
<code>-remove</code> <i>Old_Gateway_Address</i>	Specifies the old default gateway address to remove.

Exit Status

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

Examples

1. To changes the current network address and mask to the new settings, type:
`chtcpip -interface en0 -inetaddr 9.1.1.1 -netmask 255.255.255.0`
2. To changes the default gateway from 9.1.2.3 to 9.2.3.4, type:
`chtcpip -interface en0 -gateway -add 9.2.3.4 -remove 9.1.2.3`

Note: If more than one default gateway is defined, an error will occur. Only one default gateway can be defined by the `mktcpip` command.

Related Information

The `topas` command, and the `mktcpip` command.

chuser command

Purpose

Changes user attributes.

Syntax

```
chuser [-ldap] -attr Attribute=Value ... Name
```

Description

The `chuser` command changes attributes for the user identified by the *Name* variable. To change an attribute, specify the attribute name and the new value with the `-attr Attribute=Value` variable.

Use the `-ldap` flag if the user is an LDAP user. Specifying the `-ldap` flag authenticates through the LDAP load module and changes the users attributes specified by the `-attr` flag.

Attributes

The prime administrator (padmin) user can set the following attributes:

account_locked	Indicates whether the user account is locked. Possible values are as follows: true yes always The user's account is locked. The values true , yes , and always are equivalent. The user is denied access to the system. false no never The user's account is not locked. The values false , no , and never are equivalent. The user is allowed access to the system. The default value is false.
expires	Identifies the expiration date of the account. The <i>Value</i> variable is a 10-character string in the <i>MMDDhhmmyy</i> form, where <i>MM</i> = month, <i>DD</i> = day, <i>hh</i> = hour, <i>mm</i> = minute, and <i>yy</i> = last 2 digits of the years 1939 through 2038. All characters are numeric. If the <i>Value</i> variable is 0, the account does not expire. The default is 0.
histexpire	Defines the period of time (in weeks) that a user cannot reuse a password. The value is a decimal integer string. The default is 0, indicating that no time limit is set.
histsize	Defines the number of previous passwords a user cannot reuse. The value is a decimal integer string. The default is 0. Only an administrative user can change this attribute.
loginretries	Defines the number of unsuccessful login attempts allowed after the last successful login before the system locks the account. The value is a decimal integer string. A zero or negative value indicates that no limit exists. Once the user's account is locked, the user is not able to log in until the prime administrator resets the user's account_locked attribute.
maxage	Defines the maximum age (in weeks) of a password. The password must be changed by this time. The value is a decimal integer string. The default is a value of 0, indicating no maximum age. The value can be from 0 to 52.
maxexpired	Defines the maximum time (in weeks) beyond the maxage value that a user can change an expired password. The value is a decimal integer string. The default is -1, indicating restriction is set. If the maxexpired attribute is 0, the password expires when the maxage value is met. If the maxage attribute is 0, the maxexpired attribute is ignored. The value can be from 0 to 52.
maxrepeats	Defines the maximum number of times a character can be repeated in a new password. Since a value of 0 is meaningless, the default value of 8 indicates that there is no maximum number. The value is a decimal integer string. The value can be from 0 to 8.
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.
minalpha	Specifies the minimum number of alphabetic characters.
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.
minlen	Defines the minimum length of a password. The value is a decimal integer string. The default is a value of 0, indicating no minimum length. The maximum value allowed is 8. This attribute is determined by minlen or ' minalpha + minother ', whichever is greater. The values for ' minalpha + minother ' cannot be greater than 8. If ' minalpha + minother ' is greater than 8, then the effective value for minother is reduced to ' 8 - minalpha '.
minother	Defines the minimum number of non-alphabetic characters that must be in a new password. The value is a decimal integer string. The default is a value of 0, indicating no minimum number. The value can be from 0 to 8.
pgrp	Defines the Primary Group and Groups membership. The valid entries are staff and view . If this attribute is not defined, the default staff is used.
pwdwarntime	Defines the number of days before the system issues a warning that a password change is required. The value is a decimal integer string. A zero or negative value indicates that no message is issued. The value must be less than the difference of the maxage and minage attributes. Values greater than this difference are ignored and a message is issued when the minage value is reached.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Security

This command can only be run by the prime administrator (padmin) user.

Examples

To change the expiration date for the **davis** user account to 8 a.m., 1 May, 2010, type:

```
chuser -attr expires=0501080010 davis
```

Related Information

The `lsuser` command, the `mkuser` command, the `rmuser` command, and the `passwd` command.

IVM `chvet` command

Purpose

Activates Virtualization Engine™ systems technologies

Syntax

```
chvet -o <e> -k <activation code> [-m <managed system>] [--help ]
```

Description

The `chvet` command performs activation of Virtualization Engine systems technologies on the managed-system.

Flags

-o	The operation to perform. The only valid value is e to enter an activation code.
-m <i>managed system</i>	The name of the managed system for which the Virtualization Engine systems technologies activation is to be performed. The name may either be the user-defined name for the managed system, or be in the form <i>ttt-mmm*sssssss</i> , where <i>ttt</i> is the machine type, <i>mmm</i> is the model, and <i>sssssss</i> is the serial number of the managed system.
-k	The activation code (key) to enter. Letters may be entered in either upper case or lower case.
--help	Display the help text for this command and exit.

Exit Status

This command has a return code of 0 on success.

Examples

To enter an activation code, type the following command:

```
chvet -m mySystem -o e -k AlphaNumericString12345
```

chvg command

Purpose

Sets the characteristics of a volume group.

Syntax

```
chvg [ -unlock ] [ -suspend | -resume ] [-factor <num>VolumeGroup
```

Description

The **chvg** command changes the characteristics of a volume group.

Flags

-resume	Resumes normal I/O operations for a volume group.
-suspend	Drains I/O's for this volume group and suspends future I/O's.
-unlock	Unlocks the volume group. This option is provided if the volume group is left in a locked state by abnormal termination of another LVM operation (such as the command core dumping, or the system crashing). Note: Before using the -unlock flag, make sure that the volume group is not being used by another LVM command.
factor	Changes the limit of the number of physical partitions per physical volume, specified by factor. factor should be between 1-16 for 32 disk volume groups and 1-64 for 128 disk volume groups. If factor is not supplied, it is set to the lowest value such that the number of physical partitions in the volume group is less than factor x1016. If factor is specified, the maximum number of physical partitions per physical volume for the volume group changes to factor x1016.

Exit Status

See "Exit status for Virtual I/O Server commands" on page 1.

Examples

1. To suspend the volume group **vg03**, type:
chvg
-suspend vg03
2. To resume the volume group **vg03**, type:
chvg
-resume vg03
3. To change the number of physical partitions per physical volume by 4,type:
chvg -factor 4 testvg

Related Information

The **mkvg** command, the **lsvg** command, the **extendvg** command, the **reducevg** command, the **mirrorios** command, the **unmirrorios** command, the **activatevg** command, the **deactivatevg** command, the **importvg** command, the **exportvg** command, and the **syncvg** command.

chvopt command

Purpose

Change the characteristics of a virtual optical media disk within the Virtual Media Repository.

Syntax

```
chvopt -name FileName {-access Value | -mv NewName}
```

Description

The **chvopt** command renames or changes the access permission's of a virtual optical media disk within the Virtual Media Repository. If the **-access** flag is specified, then the permissions can be set to read-only (ro) or read-write (rw). If the **-mv** flag is specified the media file name is changed. The file name cannot be changed if the media file is loaded into a virtual optical device.

Flags

-dev <i>Value</i>	Specifies the new access permission. Valid values are: ro read-only rw read-write
-name <i>FileName</i>	Specifies the file name whose access permission is to be changed.
-mv <i>NewName</i>	Specifies the new file name.

Examples

To change the access permissions on the file clientData to read-only, type the following command:

```
chvopt -name clientData -access ro
```

cl_snmp command

Purpose

Issues Simple Network Management Protocol (SNMP) requests to agents and processes SNMP responses returned by agents.

Syntax

```
cl_snmp [-com][-debug Level] [-host TargetHost] [-timeout TimeoutValue] [-retry RetryNumber] [-max MaxRepetitions] [-file ConfigurationFile] [-port PortNumber] [-verbose] [-non NonRepeaters] Function [MIBvariable][VariableType][Value][...]
```

Description

Use the **cl_snmp** command to issue SNMP requests to agents and to process SNMP responses returned by agents. The Virtual I/O Server **cl_snmp** command can be used for SNMPv1, SNMPv2c, and SNMPv3 requests.

Flags

- com** Specifies the community name used to access the specified variables at the destination SNMP agent. If you do not specify a community name, the default name is public. Community names are not required when using the user-based security model.
Note: Community names are case sensitive.
- debug** *Level* Specifies a debugging level during run time. The default is 0. The higher the debugging level, the greater the number of messages that are displayed. The levels can be from 0 through 4.
- host** *TargetHost* Specifies the target host to which you want to send a request. This can be an Internet protocol address, a host name, or a winSNMP name in the clsnmp.conf configuration file. If you do not specify a host, the default is the local host.
- timeout** *TimeoutValue* Specifies the amount of time (in seconds) that the **cl_snmp** command waits for a reply from the SNMP agent. The default is 3.
- retry** *RetryNumber* Specifies the maximum number of times to retry the command if it timed out. The default is 2.
- max** *MaxRepetitions* Specifies the number of lexicographic successors to be returned for each variable binding pair after the first *-non number* successors. The parameter applies only to the **getbulk** request. This is ignored if the function request is not a **getbulk** request. For example, starting with successor *-non number+1*, return *-max number* of successors for each variable binding pair. The default is 10.
- file** *ConfigurationFile* Specifies the full path and file name of the configuration file.
- port** *PortNumber* Specifies the number of the port that listens for traps. If a port number is not specified, the **cl_snmp** trap function listens on the well-known port 162, which is the default port for **cl_snmp** traps.
- verbose** Specifies that the output from a request should be displayed using verbose output, for example, using the textual name instead of the MIB object identifier.
- non** *NonRepeaters* Specifies the number of variable binding pairs (name/value), starting with the first, for which only a single successor is returned. This parameter applies only to **getbulk** requests. This is ignored if the function request is not a **getbulk** request. The default is 0.

Function [*MIBvariable*]
[*VariableType*] [*Value*] [...]]

Specifies the SNMP function or operation to perform, which is one of the following:

- get
- getnext
- getbulk
- set
- walk
- trap
- findname

MIBVariable

Specifies the Management Information Base (MIB) object, using its object descriptor (textual name), object identifier in ASN.1 notation, or a combination of the two. When used with the walk function, this is the MIB object prefix. A prefix can be any leading portion of the complete object identifier. When used with the **findname** command, this object identifier is in the ASN.1 notation.

Value Specifies the value to be set by the SET function. If white space is needed in the value, enclose the value in quotation marks. To set a variable to a value that is also a type, you must specify the type.

VariableType

Specifies the type of value being set. To complete an SNMP SET request, the SMI_type must be known. If no type is specified, **cl_snmp** command searches first the /etc/mib.defs file and then the compiled MIB to determine the type. If the variable is not found, an error is returned. If a *VariableType* is specified, the *VariableType* takes precedence over any type that may be assigned in the MIB. The *VariableType* and value must be compatible. For example, if you specify a type of *number* and a value of *foo*, an error is returned because *foo* is not a number. *VariableType* is not case sensitive. Valid variable types are:

- bitstring
- counter
- counter32
- counter64
- display or displaystring
- gauge
- gauge32
- integer
- integer32
- ipaddress
- nsapaddress
- null
- objectidentifier or OID
- octetstring
- opaque
- opaqueascii
- timeticks
- uinteger

Request Types

findname	Sends a request that a search be done to obtain the textual name, for a given <i>MIBVariable</i> input, whose internal ASN.1 value best matches the input ASN.1 value. The search first checks the <code>/etc/mib.defs</code> file, and if a matching textual name is not found, it continues with the compiled management information base (MIB). Only one <i>MIBVariable</i> is allowed per <code>cl_snmp findname</code> invocation.
get	Sends a request to an SNMP agent for a specific MIB variable. The <code>cl_snmp</code> command then waits for a response or times out.
getbulk	Obtains the value of the variables in the MIB tree specified by the object identifier (OID) or MIB variable name. A single <code>getbulk</code> performs the same function as a series of <code>getnext</code> commands, with fewer data exchanges between the <code>cl_snmp</code> command and the SNMP agent.
getnext	Sends a request to an SNMP agent for the next MIB variable that lexicographically follows the <i>MIBVariable</i> specified. The <code>cl_snmp</code> command then waits for a response or times out.
set	Sends a request to an SNMP agent to set a specific <i>MIBVariable</i> . The <code>cl_snmp</code> command then waits for a response or times out.
trap	Listens for SNMP traps and displays trap information when they occur. Uses the default, well-known port 162 or the port number specified on the <code>-port</code> option. The <code>cl_snmp trap</code> function continues to listen for traps until the process is killed or canceled.
walk	Issues a <code>getnext</code> request for a specified prefix, then continues to issue <code>getnext</code> requests for as long as there are variables that match the specified prefix. A prefix can be any leading portion of the complete object identifier.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

To issue an SNMP request, run the following command:

```
cl_snmp -host hostname get sysName.0
```

Related Information

The `snmp_info` command and `snmp_trap` command.

clear command

Purpose

Clears the terminal screen.

Syntax

```
clear
```

Description

The `clear` command clears your screen, if possible. The `clear` command first checks the `TERM` environment variable for the terminal type. Next, the `/usr/share/lib/terminfo` directory, which contains terminal definition files, is checked to determine how to clear the screen. If the `TERM` environment variable is not set, the `clear` command exits without taking any action.

Examples

To clear your screen, enter:

```
clear
```

Files

`/usr/share/lib/terminfo`

Contains terminal information database.

cp command

Purpose

Copies files.

Syntax

To Copy a File to another File

```
cp [ -E{force|ignore|warn} ] [ -f ] [ -h ] [ -i ] [ -p ] [ -I ] [ -U ] [ - ] SourceFile TargetFile
```

To Copy a File to a Directory

```
cp [ -E{force|ignore|warn} ] [ -f ] [ -h ] [ -i ] [ -p ] [ -r | -R ] [ -H | -L | -P ] [ -I ] [ -U ] [ - ] SourceFile ... TargetDirectory
```

To Copy a Directory to a Directory

```
cp [ -E{force|ignore|warn} ] [ -f ] [ -h ] [ -i ] [ -p ] { -r | -R } [ -H | -L | -P ] [ -I ] [ -U ] [ - ] SourceDirectory ... TargetDirectory
```

Description

The **cp** command copies the source file specified by the *SourceFile* parameter to the destination file specified by the *TargetFile* parameter. If the target file exists, **cp** overwrites the contents, but the mode, owner, and group associated with it are not changed. The last access time of the *SourceFile* and the last modification time of the *TargetFile* are set to the time the copy was done. If the *TargetFile* does not exist, **cp** creates a new file named *TargetFile* that has the same mode as the source file except that the sticky bit is not set unless it was done by a superuser; the owner and group of the *TargetFile* is that of the user. When the *TargetFile* is a link to another file, **cp** overwrites the destination link with the content of the source file; the links from the *TargetFile* remains. Also, the **cp** command can copy the source files specified by the *SourceFile* parameter (or directories named by the *SourceDirectory* parameter) to the directory specified by the *TargetDirectory* parameter.

Note: If one of the source parameters is a directory, you need to specify one of the **-r** or **-R** flags.

If any directories are created by the **cp** command during the copying process, the newly created directory will have the same mode as the corresponding source directory.

You can also copy special device files. The preferred option for accomplishing this is the **-R** flag. Specifying **-R** causes the special files to be re-created under the new path name. Specifying the **-r** flag causes the **cp** command to attempt to copy the special file to a regular file.

Flags

- E** The **-E** option requires one of the following arguments. If you omit the **-E** option, **warn** is the default behavior.
- force** Fails the **cp** operation on a file if the fixed extent size or space reservation of the file cannot be preserved.
- ignore** Ignores any errors in preserving extent attributes.
- warn** Issues a warning if the space reservation or the fixed extent size of the file cannot be preserved.
- f** Specifies removal of the target file if it cannot be opened for write operations. The removal precedes any copying performed by the **cp** command.
- h** Forces the **cp** command to copy symbolic links. The default is to follow symbolic links, that is, to copy files to which symbolic links point.
- H** Take actions based on the type and contents of the file referenced by any symbolic link specified as a *SourceFile* operand.
- i** Prompts you with the name of a file to be overwritten. This occurs if the *TargetDirectory* or *TargetFile* parameter contains a file with the same name as a file specified in the *SourceFile* or *SourceDirectory* parameter. If you enter *y* or the locale's equivalent of *y*, the **cp** command continues. Any other answer prevents the **cp** command from overwriting the file.
- I** Suppresses the warning message during ACL conversion.
- L** Take actions based on the type and contents of the file referenced by any symbolic link specified as a *SourceFile* operand or any symbolic links encountered during traversal of a file hierarchy.
- p** Duplicates the following characteristics of each *SourceFile*/*SourceDirectory* in the corresponding *TargetFile* or *TargetDirectory*:
- The time of the last data modification and the time of the last access. If this duplication fails for any reason, the **cp** command will write a diagnostic message to standard error.
 - The user ID and group ID. If this duplication fails for any reason, the **cp** command may write a diagnostic message to standard error.
 - The file permission bits and the *S_ISUID* and *S_ISGID* bits. If this duplication fails for any reason, the **cp** command will write a diagnostic message to standard error.
- If the user ID or group ID cannot be duplicated, the file permission bits *S_ISUID* and *S_ISGID* are cleared.
- In order to preserve the owner ID and group ID, permission modes, modification and access times, user must have the appropriate file access permissions (user should be a superuser or have the same owner ID as the destination file)
- The target file will not be deleted if these characteristics cannot be preserved.
- Access control lists (ACLs) associated with the *SourceFile* are preserved if the target file system supports the same. If the source file contains NFS4 ACL and the target file system does not support NFS4 ACL, the NFS4 ACL is converted to AIXC.
- When ACL conversion succeeds, a warning message is printed out the *stderr*.
- P** Take actions on any symbolic link specified as a *SourceFile* operand or any symbolic link encountered during traversal of a file hierarchy.
- r** Copies file hierarchies under the file or directory specified by the *SourceFile* or *SourceDirectory* parameter (recursive copy). The **-r** flag processes special files in the same manner as regular files.

- R** Copies file hierarchies under the regular files and directories from the directory specified by the *SourceFile* or *SourceDirectory* parameter to the directory specified by the *TargetDirectory* parameter. Special file types, such as first-in, first-out (FIFO) files and block and character device files, are re-created instead of copied. Symbolic links are followed unless the **-h** flag is specified. (The **-R** flag is preferred to the **-r** flag.)
- If none of the **-H**, **-L**, or **-P** options were specified, it is unspecified which of those options will be used as the default. Consider the following:
- If the **-H** option was specified, the **cp** command will take action based on the type and contents of the file referenced by any symbolic link specified as a *SourceFile* operand.
 - If the **-L** option was specified, the **cp** command will take action based on the type and contents of the file referenced by any symbolic link specified as a *SourceFile* operand or any symbolic links encountered during traversal of a file hierarchy.
 - If the **-P** option was specified, the **cp** command will copy any symbolic link specified as a *SourceFile* operand and any symbolic links encountered during traversal of a file hierarchy and will not follow any symbolic links.
- U** Copies Extended Attributes (EA), Access Control Lists (ACL) in the *SourceFile* to the *TargetFile*. If the EA is not supported on the target file system then it is ignored. If the source ACL type is not supported on the target file system then it is converted to the compatible ACL type supported by the target file system.
- Indicates that parameters following the **--** (dash, dash) flag are to be interpreted as file names. This null flag allows the specification of file names that start with a - (minus sign).

Exit Status

This command returns the following exit values:

0	All files were copied successfully.
>0	An error occurred.

Examples

1. To make a copy of a file in the current directory, enter:

```
cp prog.c prog.bak
```

This copies `prog.c` to `prog.bak`. If the `prog.bak` file does not already exist, the **cp** command creates it. If it does exist, the **cp** command replaces it with a copy of the `prog.c` file.

2. To copy a file in your current directory into another directory, enter:

```
cp jones /home/nick/clients
```

This copies the `jones` file to `/home/nick/clients/jones`.

3. To copy a file to a new file and preserve the modification date, time, and access control list associated with the source file, enter:

```
cp -p smith smith.jr
```

This copies the `smith` file to the `smith.jr` file. Instead of creating the file with the current date and time stamp, the system gives the `smith.jr` file the same date and time as the `smith` file. The `smith.jr` file also inherits the `smith` file's access control protection.

4. To copy all the files in a directory to a new directory, enter:

```
cp /home/janet/clients/* /home/nick/customers
```

This copies only the files in the `clients` directory to the `customers` directory.

5. To copy a directory, including all its files and subdirectories, to another directory, enter:

```
cp -R /home/nick/clients /home/nick/customers
```

Note: A directory cannot be copied into itself.

This copies the `clients` directory, including all its files, subdirectories, and the files in those subdirectories, to the `customers/clients` directory.

6. To copy a specific set of files to another directory, enter:

```
cp jones lewis smith /home/nick/clients
```

This copies the `jones`, `lewis`, and `smith` files in your current working directory to the `/home/nick/clients` directory.

7. To use pattern-matching characters to copy files, enter:

```
cp programs/*.c .
```

This copies the files in the `programs` directory that end with `.c` to the current directory, signified by the single `.` (dot). You must type a space between the `c` and the final dot.

8. To copy a file to a new file and preserve the ACL and EA associated with the source file, enter:

```
cp -U smith smith.jr
```

Files

`/usr/bin/cp`

Contains the `cp` command.

Related Information

The `mv` command.

cplv command

Purpose

Copies the contents of a logical volume to a new logical volume.

Syntax

To Copy to a New Logical Volume

```
cplv [ -vg VolumeGroup ] [ -lv NewLogicalVolume | -prefix Prefix ] SourceLogicalVolume
```

To Copy to an Existing Logical Volume

```
cplv [ -f ] SourceLogicalVolume DestinationLogicalVolume
```

Description

Attention: Do not copy from a larger logical volume containing data to a smaller one. Doing so results in a corrupted file system because some data is not copied.

The `cplv` command copies the contents of *SourceLogicalVolume* to a new or existing logical volume. The *SourceLogicalVolume* parameter can be a logical volume name or a logical volume ID. The `cplv` command creates a new logical volume with a system-generated name by using the default syntax. The system-generated name is displayed.

Note: The `cplv` command cannot copy logical volumes which are in the open state, including logical volumes that are being used as backing devices for virtual storage.

Flags

-f	Copies to an existing logical volume without requesting user confirmation.
-lv <i>NewLogicalVolume</i>	Specifies the name to use, in place of a system-generated name, for the new logical volume. Logical volume names must be unique systemwide names, and can range from 1 to 15 characters.
-prefix <i>Prefix</i>	Specifies a prefix to use in building a system-generated name for the new logical volume. The prefix must be less than or equal to 13 characters. A name cannot be a name already used by another device.
-vg <i>VolumeGroup</i>	Specifies the volume group where the new logical volume resides. If this is not specified, the new logical volume resides in the same volume group as the <i>SourceLogicalVolume</i> .

Examples

1. To copy the contents of logical volume fslv03 to a new logical volume, type:

```
cp1v fslv03
```

The new logical volume is created, placed in the same volume group as fslv03, and named by the system.

2. To copy the contents of logical volume fslv03 to a new logical volume in volume group vg02, type:

```
cp1v -vg vg02 fslv03
```

The new logical volume is created, named, and added to volume group vg02.

3. To copy the contents of logical volume lv02 to a smaller, existing logical volume, lvtest, without requiring user confirmation, type:

```
cp1v -f lv02 lvtest
```

cpvdi command

Purpose

Copies a block device.

Syntax

```
cpvdi -src input_disk_image -dst output_disk_image [-isp input_storage_pool] [-osp output_storage_pool]  
[-progress] [-overwrite] [-unconfigure] [-f]
```

Description

The **cpvdi** command copies a block device image, which can be either a logical or physical volume, a file-backed device, or a file on another existing disk. The parameters of the **-src** and **-dst** flags should be fully qualified names when specifying a file or device. Specifying the **-progress** flag prints a number sign (#) on the **stderr** for each block (1 M) of data copied on to the destination.

Note: The *output_disk_image* device that you specify with the **-dst** flag cannot be a physical volume that is assigned to a shared memory pool (to be used as a paging space device by a shared memory partition).

Flags

-src <i>input_disk_image</i>	Specifies the input backing device. The input backing device might either be a physical or logical volume or a file is located in the caller's home directory, any of its subdirectories or in the /mnt directory. The root user will not have any file location restrictions.
-------------------------------------	--

-dst <i>output_disk_image</i>	Specifies the output backing device. The output backing device might either be a physical or logical volume or a file is located in the caller's home directory, any of its subdirectories or in the /mnt directory. The root user will not have any file location restrictions.
-f	Overwrites the destination file. However, if the source or destination file is configured as backing device, it copies the file without unconfiguring the device or devices. If the -unconfigure flag is used in addition to the -f flag, then the -unconfigure flag takes priority.
-progress	Specifies the process status of the command will be expressed by printing a number sign (#) to stderr for every 1024 KB block of data that has been copied.
-isp <i>input_storage_pool</i>	Specifies the name of the storage pool where the input file-backed device exists. Required if the argument of the -src flag is a file-backed device.
-osp <i>output_storage_pool</i>	Specifies the name of the storage pool where the input file-backed device exists. Required if the argument of the -dst flag is a file-backed device.
-overwrite	Overwrites the destination file, if the file is present. If you do not use this flag, an error message displays with a non-zero exit code.
-unconfigure	Specifies to unconfigure the virtual target device (VTD) and reconfigure the VTD after a copy operation. Only use this flag if the source or destination file is configured as backing device. If you do not use this flag, an error message displays with a non-zero exit code.

Exit Status

See "Exit status for Virtual I/O Server commands" on page 1.

Examples

1. To copy a physical volume disk image to a file, type the following command:

```
cpvdi -src hdisk1 -dst /home/my_home/image1_file
```
2. To copy a disk image that is a file to a physical volume, type the following command:

```
cpvdi -src /home/my_home/image2_file -dst hdisk2
```
3. To output a hash symbol for every block of data that is copied, type the following command:

```
cpvdi -src hdisk1 -dst hdisk2 -progress
```
4. To copy a file-backed device to another file-backed device, type the following command:

```
cpvdi -src image1_file -dst image2_file -isp sp01 -osp sp02
```
5. To copy a file1 to file2, when file2 already exists, type the following command:

```
cpvdi -src /home/padmin/file1 -dst /home/padmin/file2 -overwrite
```
6. To copy a file-backed device fbd1 as fbd2, when fbd1 is configured as the backing device, type the following command:

```
cpvdi -src fbd1 -dst fbd2 -isp sp01 -osp sp01 -f
```

Related Information

The **cp** command and the **backupios** command.

crontab command

Purpose

Submits, edits, lists, or removes cron jobs.

Syntax

```
crontab [ -e [UserName] | -l [UserName] | -r [UserName] | -v [UserName] | File ]
```


Description

The **crontab** command submits, edits, lists, or removes cron jobs. A cron job is a command run by the **cron** daemon at regularly scheduled intervals. To submit a cron job, specify the **crontab** command with the **-e** flag. The **crontab** command invokes an editing session that allows you to create a **crontab** file. You create entries for each cron job in this file. Each entry must be in a form acceptable to the **cron** daemon. For information on creating entries, see the **crontab** File Entry Format section.

When you finish creating entries and exit the file, the **crontab** command copies it into the **/var/spool/cron/crontabs** directory and places it in a file named for your current user name. If a file with your name already exists in the **crontabs** directory, the **crontab** command overwrites it.

Alternatively, you can create a **crontab** file by specifying the *File* parameter. If the file exists, it must be in the format the **cron** daemon expects. If the file does not exist, the **crontab** command invokes the editor. If the **EDITOR** environment variable exists, the command invokes the editor it specifies. Otherwise, the **crontab** command uses the **vi** editor.

To list the contents of your **crontab** file, specify the **crontab** command with the **-l** flag. To remove an existing file, use the **-r** flag.

The optional *UserName* parameter can be used by the owner of the **crontab** file or by the root user to edit, list, remove, or verify the status of the cron jobs for the specified user. If the *UserName* is invalid, an error message is generated and the program exits.

If the optional *UserName* parameter is not specified, the **crontab** flags are available for the root user and the current user.

Security

Only the root user or the owner of the **crontab** file can use *UserName* following the **-e**, **-l**, **-r**, and **-v** flags to edit, list, remove, or verify the **crontab** file of the specified user.

The cron Daemon

The **cron** daemon runs commands according to the **crontab** file entries. Unless you redirect the output of a cron job to standard output or error, the **cron** daemon mails you any command output or errors. If you specify a cron job incorrectly in your **crontab** file, the **cron** daemon does not run the job.

The **cron** daemon examines **crontab** files only when the **cron** daemon is initialized. When you make changes to your **crontab** file using the **crontab** command, a message indicating the change is sent to the **cron** daemon. This eliminates the overhead of checking for new or changed files at regularly scheduled intervals.

Controls on Using the crontab Command

The **/var/adm/cron/cron.allow** and **/var/adm/cron/cron.deny** files control which users can use the **crontab** command. A root user can create, edit, or delete these files. Entries in these files are user login names with one name to a line. If your login ID is associated with more than one login name, the **crontab** command uses the first login name that is in the **/etc/passwd** file, regardless of which login name you might actually be using. Also, to allow users to start **cron** jobs, the daemon attribute in the **/etc/security/user** file should be set to **TRUE**, using the **chuser** command.

The following is an example of an **cron.allow** file:

```
root
nick
dee
sarah
```

If the **cron.allow** file exists, only users whose login names appear in it can use the **crontab** command. The root user's log name must appear in the **cron.allow** file if the file exists. A system administrator can explicitly stop a user from using the **crontab** command by listing the user's login name in the **cron.deny** file. If only the **cron.deny** file exists, any user whose name does not appear in the file can use the **crontab** command.

A user cannot use the **crontab** command if one of the following is true:

- The **cron.allow** file and the **cron.deny** file do not exist (allows root user only).
- The **cron.allow** file exists but the user's login name is not listed in it.
- The **cron.deny** file exists and the user's login name is listed in it.

If neither the **cron.allow** nor the **cron.deny** file exists, only someone with root user authority can submit a job with the **crontab** command.

The crontab File Entry Format

A **crontab** file contains entries for each cron job. Entries are separated by newline characters. Each **crontab** file entry contains six fields separated by spaces or tabs in the following form:

```
minute hour day_of_month month weekday command
```

These fields accept the following values:

minute	0 through 59
hour	0 through 23
day_of_month	1 through 31
month	1 through 12
weekday	0 through 6 for Sunday through Saturday
command	a shell command

You must specify a value for each field. Except for the *command* field, these fields can contain the following:

- A number in the specified range. To run a command in May, specify 5 in the **month** field.
- Two numbers separated by a dash to indicate an inclusive range. To run a **cron** job on Tuesday through Friday, place 2-5 in the **weekday** field.
- A list of numbers separated by commas. To run a command on the first and last day of January, you would specify 1,31 in the **day_of_month** field.
- An * (asterisk), meaning all allowed values. To run a job every hour, specify an asterisk in the hour field.

Note: Any character preceded by a backslash (including the %) causes that character to be treated literally. The specification of days may be made by two fields (day of the month and day of the week). If you specify both as a list of elements, both are adhered to. For example, the following entry:

```
0 0 1,15 * 1 command
```

would run command on the first and fifteenth days of each month, as well as every Monday. To specify days by only one field, the other field should contain an * .

Specifying Commands

The **cron** daemon runs the command named in the sixth field at the selected date and time. If you include a % (percent sign) in the sixth field, the **cron** daemon treats everything that precedes it as the command invocation and makes all that follows it available to standard input, unless you escape the percent sign (\%). Blank lines and lines whose first non-blank character is the number sign (#) will be ignored. If the arguments to the command have a backslash ('\'), the backslash should be preceded by another backslash.

Note: The shell runs only the first line of the command field. All other lines are made available to the command as standard input.

The **cron** daemon starts a subshell from your **HOME** directory. If you schedule a command to run when you are not logged in and you want commands in your **.profile** file to run, the command must explicitly read your **.profile** file.

The **cron** daemon supplies a default environment for every shell, defining **HOME**, **LOGNAME**, **SHELL** (**=/usr/bin/sh**), and **PATH** (**=/usr/bin**).

Flags

-e <i>UserName</i>	Edits a copy of the user's crontab file or creates an empty file to edit if the crontab file does not exist for a valid <i>UserName</i> . When editing is complete, the file is copied into the crontab directory as the user's crontab file.
-l <i>UserName</i>	Lists the user's crontab file.
-r <i>UserName</i>	Removes the user's crontab file from the crontab directory.
-v <i>UserName</i>	Lists the status of the user's cron jobs.

Security

Auditing Events: If the auditing subsystem has been properly configured and is enabled, the **crontab** command generates the following audit record (event) every time the command is run:

Event	Information
CRON_JobRemove	Lists which users removed a cron job and when.
CRON_JobAdd	Lists which users added a cron job and when.

Exit Status

This command returns the following exit values:

0	Successful completion.
>0	An error occurred.

Examples

1. To copy a file called **mycronjobs** into the **/var/spool/cron/crontabs** directory, enter the following:

```
crontab mycronjobs
```

The file will be copied as:

```
/var/spool/cron/crontabs/<username>
```

where **<username>** is your current user name.

2. To write the time to the console every hour on the hour, enter:

```
0 * * * * echo The hour is `date` .
>/dev/console
```

3. To run the **calendar** command at 6:30 a.m. every Monday, Wednesday, and Friday, enter:

```
30 6 * * 1,3,5 /usr/bin/calendar
```

4. To run the **calendar** command every day of the year at 6:30, enter the following:

```
30 6 * * * /usr/bin/calendar
```

5. To run a script called maintenance every day at midnight in August, enter the following:

```
0 0 * 8 * /u/harry/bin/maintenance
```

6. To define text for the standard input to a command, enter:

```
0 16 * 12 5 /usr/sbin/wall%HAPPY HOLIDAY!%Remember to
turn in your time card.
```

The text following the % (percent sign) defines the standard input to the **wall** command as:

```
HAPPY HOLIDAY!
```

```
Remember to turn in your time card.
```

Files

/var/adm/cron/FIFO

A named pipe that sends messages to the **cron** daemon when new jobs are submitted with the **crontab** or **at** command.

/var/spool/cron/crontabs

Specifies the crontab spool area.

/var/adm/cron/cron.allow

Specifies a list of users allowed access to the **crontab** command.

/var/adm/cron/cron.deny

Specifies a list of users denied access to the **crontab** command.

date command

Purpose

Displays or sets the date or time.

Syntax

To Set the Date and Time as Root User

```
/usr/bin/date [ -n ] [ -u ] [ Date ] [ +FieldDescriptor ... ]
```

To Display the Date and Time

```
/usr/bin/date [ -u ] [ +FieldDescriptor ... ]
```

To adjust the Time in Seconds as root User

```
/usr/bin/date [ -a [ + | - ]sss[.fff] ]
```

Description

Attention: Do not change the date when the system is running with more than one user.

The **date** command writes the current date and time to standard output if called with no flags or with a flag list that begins with a + (plus sign). Otherwise, it sets the current date. Only a root user can change the date and time. The **date** command prints out the usage message on any unrecognized flags or input.

The following formats can be used when setting the date with the *Date* parameter:

- *mmddHHMM*[YYyy]
- *mmddHHMM*[yy]

The variables to the *Date* parameter are defined as follows:

<i>mm</i>	Specifies the month number.
<i>dd</i>	Specifies the number of the day in the month.
<i>HH</i>	Specifies the hour in the day (using a 24-hour clock).
<i>MM</i>	Specifies the minute number.
<i>YY</i>	Specifies the first two digits of the year. Note: If you do not specify the first two digits of the year, values in the range 69 to 99 refer to the twentieth century, 1969 to 1999 inclusive, and values in the range 00 to 68 refer to years in the twenty-first century, 2000 to 2068 inclusive.
<i>yy</i>	Specifies the last two digits of the year. Note: The date command accepts a 4 digit year as input. For example, if a four-digit year is specified, the date command tries to set the year to "YYyy" and fails for values which are out of range (less than 1970 and greater than 2037).

The current year is used as the default value when the year is not specified. The system operates in Coordinated Universal Time (CUT).

If you follow the **date** command with a + (plus sign) and a field descriptor, you can control the output of the command. You must precede each field descriptor with a % (percent sign). The system replaces the field descriptor with the specified value. Enter a literal % as %% (two percent signs). The **date** command copies any other characters to the output without change. The **date** command always ends the string with a new-line character.

Flags

-a [+ -] <i>sss</i> [: <i>fff</i>]	Slowly adjusts the time by <i>sss.fff</i> seconds (<i>fff</i> represents fractions of a second). This adjustment can be positive or negative. The system's clock will be sped up or slowed down until it has drifted by the number of seconds specified.
-n	Does not set the time globally on all machines in a local area network that have their clocks synchronized.
-u	Displays or sets the time in Coordinated Universal Time (CUT).

Field Descriptors

%a	Displays the locale's abbreviated weekday name.
%A	Displays the locale's full weekday name.
%b	Displays the locale's abbreviated month name.
%B	Displays the locale's full month name.
%c	Displays the locale's appropriate date and time representation. This is the default.
%C	Displays the first two digits of the four-digit year as a decimal number (00-99). A year is divided by 100 and truncated to an integer.
%d	Displays the day of the month as a decimal number (01-31). In a two-digit field, a 0 is used as leading space fill.
%D	Displays the date in the format equivalent to %m/%d/%y .
%e	Displays the day of the month as a decimal number (1-31). In a two-digit field, a blank space is used as leading space fill.
%h	Displays the locale's abbreviated month name (a synonym for %b).
%H	Displays the hour (24-hour clock) as a decimal number (00-23).
%I	Displays the hour (12-hour clock) as a decimal number (01-12).
%j	Displays the day of year as a decimal number (001-366).
%k	Displays the 24-hour-clock hour clock as a right-justified, space-filled number (0 to 23).

%m	Displays the month of year as a decimal number (01-12).
%M	Displays the minutes as a decimal number (00-59).
%n	Inserts a <new-line> character.
%p	Displays the locale's equivalent of either AM or PM.
%r	Displays 12-hour clock time (01-12) using the AM-PM notation; in the POSIX locale, this is equivalent to %I:%M:%S %p .
%S	Displays the seconds as a decimal number (00- 59).
%s	Displays the number of seconds since January 1, 1970, Coordinated Universal Time (CUT).
%t	Inserts a <tab> character.
%T	Displays the 24-hour clock (00-23) in the format equivalent to HH:MM:SS .
%u	Displays the weekday as a decimal number from 1-7 (Sunday = 7). Refer to the %w field descriptor.
%U	Displays week of the year(Sunday as the first day of the week) as a decimal number[00 - 53] . All days in a new year preceding the first Sunday are considered to be in week 0.
%V	Displays the week of the year as a decimal number from 01-53 (Monday is used as the first day of the week). If the week containing January 1 has four or more days in the new year, then it is considered week 01; otherwise, it is week 53 of the previous year.
%w	Displays the weekday as a decimal number from 0-6 (Sunday = 0). Refer to the %u field descriptor.
%W	Displays the week number of the year as a decimal number (00-53) counting Monday as the first day of the week.
%x	Displays the locale's appropriate date representation.
%X	Displays the locale's appropriate time representation.
%y	Displays the last two numbers of the year (00-99).
%Y	Displays the four-digit year as a decimal number.
%Z	Displays the time-zone name, or no characters if no time zone is determinable.
%%	Displays a % (percent sign) character.

Modified Field Descriptors

The **%E** and **%O** field descriptors can be modified to indicate a different format or specification. If the corresponding keyword (see the **era**, **era_year**, **era_d_fmt**, and **alt_digits** keywords) is not specified or not supported for the current locale, the unmodified field descriptor value is used.

%Ec	Displays the locale's alternative appropriate date and time representation.
%EC	Displays the name of the base year (or other time period) in the locale's alternative representation.
%Ex	Displays the locale's alternative date representation.
%EX	Displays the locale's alternative time representation.
%Ey	Displays the offset from the %EC field descriptor (year only) in the locale's alternative representation.
%EY	Displays the full alternative year representation.
%Od	Displays the day of the month using the locale's alternative numeric symbols.
%Oe	Displays the day of the month using the locale's alternative numeric symbols.
%OH	Displays the hour (24-hour clock) using the locale's alternative numeric symbols.
%OI	Displays the hour (12-hour clock) using the locale's alternative numeric symbols.
%Om	Displays the month using the locale's alternative numeric symbols.
%OM	Displays minutes using the locale's alternative numeric symbols.
%OS	Displays seconds using the locale's alternative numeric symbols.
%Ou	Displays the weekday as a number in the locale's alternative representation (Monday=1).
%OU	Displays the week number of the year using the locale's alternative numeric symbols. Sunday is considered the first day of the week.
%OV	Displays the week number of the year using the locale's alternative numeric symbols. Monday is considered the first day of the week.
%Ow	Displays the weekday as a number in the locale's alternative representation (Sunday =0).
%OW	Displays the week number of the year using the locale's alternative numeric symbols. Monday is considered the first day of the week.
%Oy	Displays the year (offset from %C) in alternative representation.

Exit Status

This command returns the following exit values:

- 0 The date was written successfully.
- >0 An error occurred.

Examples

1. To display current date and time, enter:

```
date
```

2. To set the date and time, enter:

```
date 0217142590
```

For a system using CST as its time zone, this sets the date and time to Sat Feb 17 14:25:00 CST 1990.

Note: You must have root authority to change the date and time.

3. To display the date and time in a specified format, enter:

```
date +"%r %a %d %h %y (Julian Date: %j)"
```

This displays the date shown in Example 2 as:

```
02:25:03 PM Fri 17 Feb 90 (Julian Date: 048)
```

Environment Variables

The following environment variables affect the execution of the **date** command.

LANG	Determines the locale to use 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 environment variable beginning with LC_ .
LC_CTYPE	Determines the locale for the interpretation of sequences of bytes of text data as characters (for example, single versus multibyte character in an argument).
LC_MESSAGES	Determines the language in which messages should be written.
LC_TIME	Determines the contents of date and time strings written by date .
NLSPATH	Determines the location of message catalogues for the processing of LC_MESSAGES .
TZ	Specifies the time zone in which the time and date are written, unless the -u option is specified. If the TZ variable is not set and the -u flag is not specified, an unspecified system default time zone is used.

deactivatevg command

Purpose

Deactivates a volume group.

Syntax

```
deactivatevg VolumeGroup
```

Description

The **deactivatevg** command deactivates the volume group specified by the *VolumeGroup* parameter along with its associated logical volumes. The logical volumes must first be closed. For example, if the logical volume contains a file system, it must be unmounted.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

1. To deactivate volume group **vg03**, type:
`deactivatevg vg03`

Related Information

The **mkvg** command, the **chvg** command, the **extendvg** command, the **reducevg** command, the **mirrorios** command, the **unmirrorios** command, the **lsvg** command, the **activatevg** command, the **importvg** command, the **exportvg** command, and the **syncvg** command.

IVM defsysplanres command

Purpose

Defines the system plan resource.

Syntax

```
defsysplanres -r osinstall -n <resource name> -v <resource value> [-d <resource description>] [-help]
```

Description

The **defsysplanres** command defines a system plan resource for use by system plans deployed from the Integrated Virtualization Manager (IVM).

Flags

-r Specifies the type of system plan resources to define. To define the installation resources for the operating system, the only valid value is **osinstall**.

This command does not create the resource. It sets the reference to an operating system installation resource that can later be used within system plans that are deployed from the IVM where the resource has been defined. This resource is defined only to the IVM and not to any system plan. If the resource defined appears with a location attribute in a system plan, the plan location is used instead.

-n	Specifies the name of the system plan resource you want to define. When a system plan is deployed that needs to use the resource, the name is used to retrieve the plan resource location attribute. Valid values follow: <ul style="list-style-type: none"> • <i>resource name</i> - the name of the resource to define. • <i>sles9</i> - Define the installation resources for installing the SLES9 Linux operating system with a system plan. • <i>rhel4</i> - Define the installation resources for installing the Red Hat 4 Linux operating system with a system plan. • <i>aix52</i> - Define the installation resources for installing the AIX 5.2 operating system with a system plan. • <i>aix53</i> - Define the installation resources for installing the AIX 5.3 operating system with a system plan.
-v	Specifies the value of the resource. This value is not checked for validity by the command. It is treated as a string value.
-d	Specifies a description for the resource. This is an open text description field.
-help	Displays the help text for this command and exit.

Exit Status

The following exit values are returned:

0	Successful completion.
>0	Invalid flag, argument, or command failure.

Examples

1. To define the plan resource for the **osinstall** type used as the Red Hat installation source, type:

```
defsysplanres -r osinstall -n rhel4
               -v rhel43
               -d "Red hat install source"
```
2. To define the plan resource for the **osinstall** type used as the AIX 5.3 installation source, type:

```
defsysplanres -r osinstall -n aix53
               -v aix53install
```

Related Information

The `deploysysplan`, `lssysplanres`, `lssysplan`, and `rmsysplanres` commands.

IVM deploysysplan command

Purpose

Deploys a system plan (an XML-based system configuration) generated using the **mksysplan** command or the System Planning Tool (SPT) on the managed system.

Syntax

```
deploysysplan -f <system plan file name> -o dv | v | d [-m <managed system>] [--force] [--plan <system plan name>] [-v] [-clientinstall] [--help]
```

Description

The **deploysysplan** command deploys a managed system's partition plan from an XML input file.

Flags

-f <i>SysPlanFileName</i>	Specifies the XML file (saved as a compressed stream with extension <code>.sysplan</code>) that contains the system plan that is to be deployed. This file is not changed by the deploysysplan command. The file must be in the system plan file directory (<code>/opt/hsc/data/sysplan</code>) on the management partition. The file name that you specify must end in <code>".sysplan."</code> The file name can be uppercase, lowercase, or mixed.
-o	Specifies the deployment option to be used when deploying the system plan. The options are as follows: <ul style="list-style-type: none">dv Validates and deploys the system plan on the managed system.v Validates the system plan for the managed system but does not deploy it.d Indicates that the system plan should not be validated against the managed system, but the plan-specified partitions and profiles on the managed system are to be deployed.
-m	Specifies the managed system's name where the plan is to be deployed. The system-plan file specifies the managed system's name where that system plan was intended to be deployed. Because IVM can cater to only one managed system, specify only the defaulted managed system using the -m option. This option is kept just to maintain the consistency with other IVM commands. The name can either be the user-defined name for the managed system, or be in the form <code>tttt-mmm*sssssss</code> , where <code>tttt</code> is the machine type, <code>mmm</code> is the model, and <code>sssssss</code> is the serial number of the managed system.
--force	When the managed system has existing partitions other than the Virtual I/O Server partition, that the system plan contains and that are in an active state, this option allows the deploysysplan command to proceed with deployment. Otherwise, if active partitions are found, the deploysysplan command will fail with an error message.
-v	Specifies that a list of identifiers that identify the steps of the validated system plan should be written to standard output (stdout). These steps are part of the plan but not yet implemented on the managed system. Otherwise, this list is not written out. When you specify the -dv or -d options, this parameter specifies that a list of log entries that record the steps performed as part of deploying the system plan be written to stdout. Otherwise, this list is not written out.
--plan <i><system plan name></i>	Specifies the which system plan files can contain more than one system definition. If the file contains more than one system, this flag specifies the name of the system within the system plan to deploy. If this name is not specified and there is more than one system defined, an error listing the system name is returned. If there is only one system plan in the file, this flag is ignored.
-clientinstall	Identifies the operating systems to be installed on the client logical partitions and performs an automatic installation of one or more of the clients, as defined in the system plan.
--help	Displays the help text for this command and exit.

Exit Status

See "Exit status for Virtual I/O Server commands" on page 1.

Examples

To deploy a system plan from file `mysysplan.xml` onto the managed system configured in the file. Run the following command to validate the plan against the managed system before deploying it:

```
deploysysplan -f mysysplan.sysplan -o dv
```

Deploy a system plan from file `mysysplan.xml` and specify the managed system `mySystem`. Validate the plan against the managed system before deploying it. Deploy the plan, if valid, even though there might be active client partitions on the managed system:

```
deploysysplan -f mysysplan.sysplan -o dv -m mySystem --force
```

Deploy a system plan from file *mysysplan.xml* onto a managed system by machine type, model number, and serial number of the managed system. Validate the plan against the managed system before deploying it. Write any errors found in validation or deployment to stdout:

```
deploysysplan -f mysysplan.sysplan -o dv -m 9406-570*34134888 -v
```

Validate the ability to deploy a system plan from file *mysysplan.sysplan* onto managed system *mySystem*. Validate the plan against the managed system to determine if that plan can be deployed there but do not actually deploy the plan. Write out the steps in the system plan that are part of the plan but not yet implemented to stdout. Write any errors found in validation or deployment to stdout:

```
deploysysplan -f mysysplan.sysplan -o v -m mySystem -v
```

Deploy a system plan from file *mysysplan.sysplan* onto managed system *mySystem*. Do not validate the plan against the managed system to determine if that plan can be deployed there. Write information about the steps performed in deployment to stdout. Write any errors found in validation or deployment to stdout. Deploy the plan, if valid, even though there might be active partitions on the managed system:

```
deploysysplan -f mysysplan.sysplan -o d -m mySystem -v -force
```

Deploy a system plan from file *mysysplan.sysplan* onto managed system *mySystem*. Validate the plan against the managed system before deploying it. Deploy the plan, if valid, even though there might be active partitions on the managed system. Only deploy those steps from the system plan that have been identified by the *-i* parameter:

```
deploysysplan -f mysysplan.sysplan -o dv -m mySystem -force -i "23 45 33 465 99 3 4"
```

Deploy a system plan from the *mysysplan.sysplan* file onto the managed system configured in the file. Type the following command to validate the plan against the managed system before deploying it, and install the client partitions with the operating systems specified in the system plan.

```
deploysysplan -f mysysplan.sysplan -o dv -clientinstall
```

Related Information

The **lssysplan** command, **mksysplan** command, and **rmsysplan** command.

diagmenu command

Purpose

Places the user into the diagnostic menus.

Syntax

```
diagmenu
```

Description

The **diagmenu** command performs hardware problem determination and maintenance. When the user suspects there is a problem, **diagmenu** assists in finding it. Through the diagnostic menus, users can perform tasks such as running diagnostics, hot plugging devices, formatting and certifying media, and managing RAID devices.

Once inside the diagnostic menus, task specific help can be obtained by pressing the **F1** key.

Related Information

The **errlog** command.

dsmc command

Purpose

Backups and stores data generated while using the IBM Tivoli Storage Manager Server.

Syntax

```
dsmc [ -incremental | -schedule ]
```

```
dsmc [ -query | -restore ] argv
```

Description

The **dsmc** command allows the user to back up and restore data from the IBM Tivoli Storage Manager Server.

Flag

-incremental	Backs up all new or changed files or directories in the default client domain or from file systems, directories, or files you specify, unless you exclude them from backup services.
---------------------	--

- query [argument]** Query functions to query backups on the IBM Tivoli Storage Manager Server
- access** Displays a list of current authorization rules.
 - archive** Displays a list of archived files
 - backup** Displays a list of back up versions.
 - backupset**
 - Queries a backup set from a local file, tape device, or the IBM Tivoli Storage Manager Server.
 - filespace**
 - Displays a list of file spaces in IBM Tivoli Storage Manager storage. You can also specify a single file space name to query.
 - group** Displays information about group backups and their members.
 - image** Displays information about image backups.
 - inclexcl**
 - Displays a list of include-exclude statements in the order in which they are processed during backup and archive operations.
 - mgmtclass**
 - Displays information about available management classes.
 - node** Displays all the nodes for which an administrative user ID has authority to perform operations.
 - options**
 - Displays all or part of your options and their current settings.
 - restore** Displays a list of your restartable restore sessions in the server database.
 - schedule**
 - Displays information about scheduled events for your node.
 - session** Displays information about your session, including the current node name, when the session was shed, server information, and server connection information.
 - systeminfo**
 - Gathers IBM Tivoli Storage Manager system information and outputs this information to a file or the console.
 - was** Displays backups of the WebSphere® Application Server (WAS) Network Deployment Manager (contains setup, application files, and configuration information) or the Application Server that match the node name and type of the WAS group backup that you specify.
- restore [argument]** Restores copies of backup versions of your files from an IBM Tivoli Storage Manager server.
- backupset**
 - Restores a backup set from the IBM Tivoli Storage Manager server or a local file. You can also restore a backup from a tape device.
 - group** Restores specific members or all members of a group backup.
 - image** Restores a file system or raw volume image backup.
 - nas** Restores the image of a file system belonging to a Network Attached Storage (NAS) file server.
 - was** Restores the WebSphere Application Server (WAS) Network Deployment Manager (contains setup, application files, and configuration information) or the Application Server from the Tivoli Storage Manager server.
- schedule** Starts the client scheduler on the workstation.

IBM Tivoli Storage Manager

IBM Tivoli Storage Manager can help protect computers running a variety of different operating environments, including the Virtual I/O Server, on a variety of different hardware. Configuring the IBM Tivoli Storage Manager client on the Virtual I/O Server enables you to include the Virtual I/O Server in your standard backup framework.

Attributes	Description	Value
DSMC_CONFIG	Specifies the location of the dsm.sys configuration file.	/home/padmin/tivoli/tsm/dsm.sys
DSMC_DIR	Specifies the location of the Tivoli Storage Manager configuration directory.	/home/padmin/tivoli/tsm/

Exit Status

The following exit values are returned:

0	Successful completion.
>0	An error occurred.

Examples

To backup an image to the IBM Tivoli Storage Manager server, type:

```
dsmc -incremental <mksysb_file_to_backup>
```

To view backup images on the IBM Tivoli Storage Manager server, type:

```
dsmc -query backup <mksysb_file_to_backup>
```

entstat command

Purpose

Shows Ethernet device driver and device statistics.

Syntax

```
entstat [ -all ] [ -reset ] Device_Name
```

Description

The **entstat** command displays the statistics gathered by the specified Ethernet device driver. The user can optionally specify that the device-specific statistics be displayed in addition to the device generic statistics. If no flags are specified, only the device generic statistics are displayed.

Flags

-all	Displays all the statistics, including the device-specific statistics.
-reset	Resets all the statistics back to their initial values.

Parameters

Device_name The name of the Ethernet device, for example, ent0.

Exit Status

The statistic fields displayed in the output of the **entstat** command and their descriptions are as follows.

Note: Some adapters may not support a specific statistic. The value of non-supported statistic fields is always 0.

Title Fields

Device Type	Displays the description of the adapter type.
Hardware Address	Displays the Ethernet network address currently used by the device.
Elapsed Time	Displays the real time period which has elapsed since last time the statistics were reset. Part of the statistics may be reset by the device driver during error recovery when a hardware error is detected. There will be another Elapsed Time displayed in the middle of the output when this situation has occurred in order to reflect the time differences between the statistics.

Transmit Statistics Fields

Packets	The number of packets transmitted successfully by the device.
Bytes	The number of bytes transmitted successfully by the device.
Interrupts	The number of transmit interrupts received by the driver from the adapter.
Transmit Errors	The number of output errors encountered on this device. This is a counter for unsuccessful transmissions due to hardware or network errors.
Packets Dropped	The number of packets accepted by the device driver for transmission which were not (for any reason) given to the device.
Max Packets on S/W Transmit Queue	The maximum number of outgoing packets ever queued to the software transmit queue.
S/W Transmit Queue Overflow	The number of outgoing packets which have overflowed the software transmit queue.
Current S/W+H/W Transmit Queue Length	The number of pending outgoing packets on either the software transmit queue or the hardware transmit queue.
Broadcast Packets	The number of broadcast packets transmitted without any error.
Multicast Packets	The number of multicast packets transmitted without any error.
No Carrier Sense	The number of unsuccessful transmissions due to the no carrier sense error.
DMA Underrun	The number of unsuccessful transmissions due to the DMA underrun error.
Lost CTS Errors	The number of unsuccessful transmissions due to the loss of the Clear-to-Send signal error.
Max Collision Errors	The number of unsuccessful transmissions due to too many collisions. The number of collisions encountered exceeded the number of retries on the adapter.
Late Collision Errors	The number of unsuccessful transmissions due to the late collision error.
Deferred	The number of outgoing packets deferred during transmission. Deferred means that the adapter had to defer while trying to transmit a frame. This condition occurs if the network is busy when the adapter is ready to transmit. The adapter will only defer the first attempt to send a packet. After that the adapter will transmit the packet without checking. If the network is still busy then a collision will be recorded.
SQE Test	Contains the number of "Signal Quality Error" Tests (i.e. Heartbeat) performed successfully during transmission.
Timeout Errors	The number of unsuccessful transmissions due to adapter reported timeout errors.

Single Collision Count	The number of outgoing packets with single (only one) collision encountered during transmission.
Multiple Collision Count	The number of outgoing packets with multiple (2 - 15) collisions encountered during transmission
Current® HW Transmit Queue Length	The number of outgoing packets which currently exist on the hardware transmit queue.
CRC Errors	The number of incoming packets with the Checksum (FCS) error.
DMA Overrun	The number of incoming packets with the DMA overrun error.
Alignment Errors	The number of incoming packets with the alignment error.
No Resource Errors	The number of incoming packets dropped by the hardware due to the no resource error. This error usually occurs because the receive buffers on the adapter were exhausted. Some adapters may have the size of the receive buffers as a configurable parameter. Check the device configuration attributes for possible tuning information.
Receive Collision Errors	The number of incoming packets with the collision errors during the reception.
Packet Too Short Errors	The number of incoming packets with the length error indicating that the packet size is less than the Ethernet minimum packet size.
Packet Too Long Errors	The number of incoming packets with the length error indicating that the packet size is bigger than the Ethernet maximum packet size.
Packets Discarded by Adapter	The number of incoming packets dropped by the hardware for any other reasons.
Receiver Start Count	The number of times that the receiver (receive unit) on the adapter has been started.

Examples

- To display the device generic statistics for **ent0**, type:

```
entstat ent0
```

This produces output similar to the following:

```
ETHERNET STATISTICS (ent0) :
Device Type: Ethernet High Performance LAN Adapter
Hardware Address: 02:60:8c:2e:d0:1d
Elapsed Time: 0 days 0 hours 8 minutes 41 seconds

Transmit Statistics:      Receive Statistics:
-----
Packets: 3                Packets: 2
Bytes: 272                Bytes: 146
Interrupts: 3            Interrupts: 2
Transmit Errors: 0       Receive Errors: 0
Packets Dropped: 0      Packets Dropped: 0
Max Packets on S/W      Bad Packets: 0
Transmit Queue:0
S/W Transmit Queue
Overflow: 0
Current S/W+H/W Transmit
Queue Length: 0

Broadcast Packets: 2     CRC Errors: 0
Multicast Packets: 0    Broadcast Packets: 1
No Carrier Sense: 0     Multicast Packets: 0
DMA Underrun: 0         DMA Overrun: 0
Lost CTS Errors: 0      Alignment Errors: 0
Max Collision Errors: 0 No Resource Errors: 0
Late Collision Errors: 0 Receive Collision Errors: 0
Deferred: 0             Packet Too Short Errors: 0
SQE Test: 0            Packet Too Long Errors: 0
Timeout Errors: 0      Packets Discarded by Adapter: 0
```



```
Single Collision          Receiver Start Count: 1
Count: 0
Multiple Collision Count: 0
Current HW Transmit Queue
Length: 0
```

General Statistics:

```
-----
No mbuf Errors: 0
Adapter Reset Count: 0
Driver Flags: Up Broadcast Running Simplex
```

2. To display the Ethernet device generic statistics and the Ethernet device-specific statistics for **ent0**, type:

```
entstat -all
```

Results similar to the following will be displayed:

```
ETHERNET STATISTICS (ent0) :
Device Type: Ethernet High Performance LAN Adapter
Hardware Address: 02:60:8c:2e:d0:1d
Elapsed Time: 0 days 2 hours 6 minutes 30 seconds
```

Transmit Statistics:	Receive Statistics:
-----	-----
Packets: 3	Packets: 2
Bytes: 272	Bytes: 146
Interrupts: 3	Interrupts: 2
Transmit Errors: 0	Receive Errors: 0
Packets Dropped: 0	Packets Dropped: 0
Max Packets on S/W	Receiver Start Count: 1
Transmit Queue:0	
Bad Packets: 0	
S/W Transmit Queue Overflow: 0	
Current S/W+H/W Transmit Queue Length: 0	
Broadcast Packets: 0	Broadcast Packets: 0
Multicast Packets: 0	Multicast Packets: 0
No Carrier Sense: 0	CRC Errors: 0
DMA Underrun: 0	DMA Overrun: 0
Lost CTS Errors: 0	Alignment Errors: 0
Max Collision Errors: 0	No Resource Errors: 0
Late Collision Errors: 0	Receive Collision Errors: 0
Deferred: 0	Packet Too Short Errors: 0
SQE Test: 0	Packet Too Long Errors: 0
Timeout Errors: 0	Packets Discarded by Adapter: 0
Single Collision Count: 0	Receiver Start Count: 1
Multiple Collision Count: 0	
Current HW Transmit Queue Length: 0	

General Statistics:

```
-----
No mbuf Errors: 0
Adapter Reset Count: 0
Driver Flags: Up Broadcast Running Simplex
```

Ethernet High Performance LAN Adapter Specific Statistics:

```
-----
Receive Buffer Pool Size: 37
Transmit Buffer Pool Size: 39
In Promiscuous Mode for IP Multicast: No
Packets Uploaded from Adapter: 0
Host End-of-List Encountered: 0
82586 End-of-List Encountered: 0
Receive DMA Timeouts: 0
Adapter Internal Data: 0x0 0x0 0x0 0x0 0x0
```

Related Information

The `optimizenet` command.

errlog command

Purpose

Displays or clears the error log.

Syntax

```
errlog [ -ls][ -seq SequenceNumber ] | -rm Days ]
```

Description

The `errlog` command generates an error report from entries in the Virtual I/O Server error log or deletes all entries from the error log older the number of days specified by the *Days* parameter. To delete all error-log entries, specify a value of 0 for the *Days* parameter. If no flags are specified a summary report is displayed.

Flags

<code>-ls</code>	Displays detailed information about errors in the error log file.
<code>-rm <i>Days</i></code>	Deletes error-log entries older than the number of days specified by the <i>Days</i> parameter.
<code>-seq <i>SequenceNumber</i></code>	Displays information about a specific error in the error log file by the sequence number.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

1. To display a complete detailed report, enter:
`errlog -ls`
2. To delete error-log entries older than 5 days, enter:
`errlog -rm 5`
3. To delete all error-log entries, enter:
`errlog -rm 0`

Related Information

The `diagmenu` command.

exportvg command

Purpose

Exports the definition of a volume group from a set of physical volumes.

Syntax

```
exportvg VolumeGroup
```

Description

The **exportvg** command removes the definition of the volume group specified by the *VolumeGroup* parameter from the system. The primary use of the **exportvg** command, coupled with the **importvg** command, is to allow portable volumes to be exchanged between systems. Only a complete volume group can be exported, not individual physical volumes.

Using the **exportvg** command and the **importvg** command, you can also switch ownership of data on physical volumes shared between two processors.

Mount points longer than 128 characters will not automatically be re-mounted when the volume group is imported using the **importvg** command and should not be used.

Note: A volume group that has a paging space volume on it cannot be exported.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

1. To remove volume group **vg02** from the system, type:
`exportvg vg02`

Note: The volume group must be deactivated before exporting.
The definition of **vg02** is removed from the system and the volume group cannot be accessed.

Restrictions

Mount points longer than 128 characters should not be used.

A volume group that has a paging space volume on it cannot be exported.

Related Information

The **mkvg** command, the **chvg** command, the **lsvg** command, the **reducevg** command, the **mirrorios** command, the **unmirrorios** command, the **activatevg** command, the **deactivatevg** command, the **importvg** command, the **extendvg** command, and the **syncvg** command.

extendlv command

Purpose

Increases the size of a logical volume.

Syntax

```
extendlv LogicalVolume Size [ PhysicalVolume ... ]
```

Description

The **extendlv** command increases the size of the logical volume. The *LogicalVolume* parameter can be a logical volume name or a logical volume ID. To limit the allocation to specific physical volumes, use the names of one or more physical volumes in the *PhysicalVolume* parameter; otherwise, all the physical volumes in a volume group are available for allocating new physical partitions.

The *Size* parameter specifies the minimum size the logical volume should be increased by. When specifying *Size* the following conventions must be used:

Size	Logical volume size
###M/m	### MB
###G/g	### GB

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1

Examples

1. To increase the size of the logical volume **lv05** by three megabytes, type:

```
extendlv lv05 3M
```
2. To request a logical volume named **lv05** with a minimum size of 10MB, type:

```
extendlv lv05 10M
```

The **extendlv** command will determine the number of partitions needed to create a logical volume of at least that size.

Related Information

The **mkiv** command, the **lsiv** command, and the **rmlv** command.

extendvg command

Purpose

Adds physical volumes to a volume group.

Syntax

```
extendvg [ -f ] VolumeGroup PhysicalVolume ...
```

Description

The **extendvg** command increases the size of volume group by adding one or more physical volumes.

The physical volume is checked to verify that it is not already in another volume group. If the system determines the physical volume belongs to a volume group that is activated, it exits. But if the system detects a description area from a volume group that is not activated, it prompts you for confirmation while continuing with the command. The previous contents of the physical volume are lost, so you must be cautious when using the override function.

Flags

- f** Forces the physical volume to be added to the specified volume group unless the following conditions are true:
- The physical volume is a member of another volume group in the device configuration database.
 - The physical volume is a member of the another volume group that is active.
 - The physical volume is assigned to a shared memory pool (to be used as a paging space device by a shared memory partition).

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

To add physical volumes **hdisk3** and **hdisk8** to volume group **vg3**, type the following command:

```
extendvg vg3 hdisk3 hdisk8
```

Related Information

The **lsvg** command, the **chvg** command, the **mkvg** command, the **reducevg** command, the **mirrorios** command, the **unmirrorios** command, the **activatevg** command, the **deactivatevg** command, the **importvg** command, the **exportvg** command, and the **syncvg** command.

fcstat command

Purpose

Displays the statistics gathered by the specified fibre channel device driver.

Syntax

To display statistics gathered by the specified fibre channel device driver:

```
fcstat -Device_Name
```

Description

The **fcstat** command displays the statistics gathered by the specified fibre channel device driver. It collects the statistics using the following process:

1. Opens the message catalog of **fcstat** and checks the parameter list.
2. Accesses the ODM database for information that relates to the selected adapter.
3. Accesses the ODM database for information that relates to ports of the selected adapter.
4. Opens and accesses adapter statistics.
5. Reports statistics and exits.

Flags

<i>Device_Name</i>	The name of the fibre channel device, for example, fcs0.
--------------------	--

Exit Status

The following exit values are returned:

0	Successful completion.
>0	An error occurred.

Examples

To display the statistics for fibre channel device driver *fcs0*, enter:

```
fcstat fcs0
```

fsck command

Purpose

Checks file system consistency and interactively repairs the file system.

Syntax

```
fsck [ FileSystem ... ]
```

Description

The **fsck** command checks and interactively repairs inconsistent file systems. Normally, the file system is consistent, and the **fsck** command merely reports on the number of files, used blocks, and free blocks in the file system. If the file system is inconsistent, the **fsck** command displays information about the inconsistencies found and prompts you for permission to repair them. If no *FileSystem* is specified, all file systems are checked.

The **fsck** command checks for the following inconsistencies:

- Blocks or fragments allocated to multiple files.
- i-nodes containing block or fragment numbers that overlap.
- i-nodes containing block or fragment numbers out of range.
- Discrepancies between the number of directory references to a file and the link count of the file.
- Illegally allocated blocks or fragments.
- i-nodes containing block or fragment numbers that are marked free in the disk map.
- i-nodes containing corrupt block or fragment numbers.
- A fragment that is not the last disk address in an i-node. This check does not apply to compressed file systems.
- Files larger than 32KB containing a fragment. This check does not apply to compressed file systems.
- Size checks:
 - Incorrect number of blocks.
 - Directory size not a multiple of 512 bytes.

Note: These checks do not apply to compressed file systems.

- Directory checks:
 - Directory entry containing an i-node number marked free in the i-node map.
 - i-node number out of range.
 - Dot (.) link missing or not pointing to itself.
 - Dot dot (..) link missing or not pointing to the parent directory.
 - Files that are not referenced or directories that are not reachable.
- Inconsistent disk map.
- Inconsistent i-node map.

In addition to its messages, the **fsck** command records the outcome of its checks and repairs through its exit value. This exit value can be any sum of the following conditions:

0	All checked file systems are now okay.
2	The fsck command was interrupted before it could complete checks or repairs.
4	The fsck command changed the file system; the user must restart the system immediately.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

1. To check a file system, enter:

```
fsck /dev/hd1
```

This command checks the unmounted file system located on the `/dev/hd1` device.

ftp command

Purpose

Transfers files between a local and a remote host.

Syntax

```
ftp [ -d ] [ -g ] [ -i ] [ -n ] [ -v ] [ -f ] [ -k realm ] [-q[-C]][ HostName [ Port ] ]
```

Description

The **ftp** command uses the File Transfer Protocol (FTP) to transfer files between the local host and a remote host or between two remote hosts.

The FTP protocol allows data transfer between hosts that use dissimilar file systems. Although the protocol provides a high degree of flexibility in transferring data, it does not attempt to preserve file attributes (such as the protection mode or modification times of a file) that are specific to a particular file system. Moreover, the FTP protocol makes few assumptions about the overall structure of a file system and does not provide or allow such functions as recursively copying subdirectories.

Issuing Subcommands

At the `ftp>` prompt, you can enter subcommands to perform tasks such as listing remote directories, changing the current local and remote directory, transferring multiple files in a single request, creating and removing directories, and escaping to the local shell to perform shell commands. See the Subcommands section for a description of each subcommand.

If you execute the **ftp** command and do not specify the *HostName* parameter for a remote host, the **ftp** command immediately displays the `ftp>` prompt and waits for an **ftp** subcommand. To connect to a remote host, execute the **open** subcommand. When the **ftp** command connects to the remote host, the **ftp** command then prompts for the login name and password before displaying the `ftp>` prompt again. The **ftp** command is unsuccessful if no password is defined at the remote host for the login name.

The **ftp** command interpreter, which handles all subcommands entered at the `ftp>` prompt, provides facilities that are not available with most file-transfer programs, such as:

- Handling file-name parameters to **ftp** subcommands
- Collecting a group of subcommands into a single subcommand macro
- Loading macros from a `$HOME/.netrc` file

These facilities help simplify repetitive tasks and allow you to use the **ftp** command in unattended mode.

The command interpreter handles file-name parameters according to the following rules:

- If a - (hyphen) is specified for the parameter, standard input (stdin) is used for read operations and standard output (stdout) is used for write operations.
- If the preceding check does not apply and file-name expansion is enabled (see the **-g** flag or the **glob** subcommand), the interpreter expands the file name according to the rules of the C shell. When globbing is enabled and a pattern-matching character is used in a subcommand that expects a single file name, results may be different than expected.

For example, the **append** and **put** subcommands perform file-name expansion and then use only the first file name generated. Other **ftp** subcommands, such as **cd**, **delete**, **get**, **mkdir**, **rename**, and **rmdir**, do not perform file-name expansion and take the pattern-matching characters literally.

- For the **get**, **put**, **mget**, and **mput** subcommands, the interpreter has the ability to translate and map between different local and remote file-name syntax styles (see the **case**, **ntrans**, and **nmap** subcommands) and the ability to modify a local file name if it is not unique (see the **runique** subcommand). Additionally, the **ftp** command can send instructions to a remote **ftpd** server to modify a remote file name if it is not unique (see the **sunique** subcommand).
- Use double quotation marks (" ") to specify parameters that include blank characters.

Note: The **ftp** command interpreter does not support pipes. It also does not necessarily support all multibyte-character file names.

To end an **ftp** session when you are running interactively, use the **quit** or **bye** subcommand or the End of File (Ctrl-D) key sequence at the **ftp>** prompt. To end a file transfer before it has completed, press the Interrupt key sequence. The default Interrupt key sequence is Ctrl-C. The **stty** command can be used to redefine this key sequence.

The **ftp** command normally halts transfers being sent (from the local host to the remote host) immediately. The **ftp** command halts transfers being received (from the remote host to the local host) by sending an FTP ABOR instruction to the remote FTP server and discarding all incoming file transfer packets until the remote server stops sending them. If the remote server does not support the ABOR instruction, the **ftp** command does not display the **ftp>** prompt until the remote server has sent all of the requested file. Additionally, if the remote server does something unexpected, you may need to end the local **ftp** process.

Security and Automatic Login

If Standard is the current authentication method:

The **ftp** command also handles security by sending passwords to the remote host and permits automatic login, file transfers, and logoff.

If you execute the **ftp** command and specify the host name (*HostName*) of a remote host, the **ftp** command tries to establish a connection to the specified host. If the **ftp** command connects successfully, the **ftp** command searches for a local **\$HOME/.netrc** file in your current directory or home directory. If the file exists, the **ftp** command searches the file for an entry initiating the login process and command macro definitions for the remote host. If the **\$HOME/.netrc** file or automatic login entry does not exist the **ftp** command prompts the user for a user name and password. The command displays the prompt whether or not the *HostName* parameter is specified on the command line.

Note: The queuing system does not support multibyte host names.

If the **ftp** command finds a **\$HOME/.netrc** automatic login entry for the specified host, the **ftp** command attempts to use the information in that entry to log in to the remote host. The **ftp** command also loads any command macros defined in the entry. In some cases (for example, when the required password is not listed in an automatic login entry), the **ftp** command prompts for the password before displaying the **ftp>** prompt.

Once the **ftp** command completes the automatic login, the **ftp** command executes the **init** macro if the macro is defined in the automatic login entry. If the **init** macro does not exist or does not contain a **quit** or **bye** subcommand, the **ftp** command then displays the ftp> prompt and waits for a subcommand.

Note: The remote user name specified either at the prompt or in a **\$HOME/.netrc** file must exist and have a password defined at the remote host. Otherwise, the **ftp** command fails.

If Kerberos 5 is the current authentication method:

The **ftp** command will use the extensions to ftp specifications as defined in IETF draft document "draft-ietf-cat-ftpsec-09.txt". The FTP security extensions will be implemented using the Generic Security Service API (GSSAPI) security mechanism. The GSSAPI provides services independent to the underlying security and communication mechanism. The GSSAPI is defined in rfc 1508 and 1509.

The **ftp** command will use the AUTH and ADAT commands to authenticate with the **ftpd** daemon. If both support Kerberos authentication, then they will use the local users DCE credentials to authenticate the user on the remote system. If this fails and Standard authentication is configured on both systems, the process described above will be used.

The *HostName* parameter is the name of the host machine to which files are transferred. The optional *Port* parameter specifies the ID of the port through which to transmit. (The **/etc/services** file specifies the default port.)

Flags

- C Allows the user to specify that the outgoing file sent using the **send_file** command must be cached in the Network Buffer Cache (NBC). This flag cannot be used unless the **-q** flag is specified. This flag is only applicable when a file is being sent out in the binary mode with no protection.
- d Sends debugging information about **ftp** command operations to the **syslogd** daemon. If you specify the **-d** flag, you must edit the **/etc/syslog.conf** file and add one of the following entries:
user.info FileName

OR
user.debug FileName
Note: The **syslogd** daemon debug level includes info level messages.

If you do not edit the **/etc/syslog.conf** file, no messages are produced. After changing the **/etc/syslog.conf** file, run the **refresh -s syslogd** or **kill -1 SyslogdPID** command to inform the **syslogd** daemon of the changes to its configuration file. For more information about debug levels, refer to the **/etc/syslog.conf** file. Also, refer to the **debug** subcommand.
- g Disables the expansion of metacharacters in file names. Interpreting metacharacters can be referred to as expanding (sometimes called globbing) a file name. See the **glob** subcommand.
- i Turns off interactive prompting during multiple file transfers. See the **prompt**, **mget**, **mput**, and **mdelete** subcommands for descriptions of prompting during multiple file transfers.
- n Prevents an automatic login on the initial connection. Otherwise, the **ftp** command searches for a **\$HOME/.netrc** entry that describes the login and initialization process for the remote host. See the **user** subcommand.
- q Allows the user to specify that the **send_file** subroutine must be used for sending the file on the network. This flag is only applicable when a file is being sent out in the binary mode with no protection.
- v Displays all the responses from the remote server and provides data transfer statistics. This display mode is the default when the output of the **ftp** command is to a terminal, such as the console or a display.

If stdin is not a terminal, the **ftp** command disables verbose mode unless the user invoked the **ftp** command with the **-v** flag or issued the **verbose** subcommand.
- f Causes the credentials to be forwarded. This flag will be ignored if Kerberos 5 is not the current authentication method.

-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.

Subcommands

The following **ftp** subcommands can be entered at the **ftp>** prompt. Use double quotation marks(" ") to specify parameters that include blank characters.

! <i>[Command [Parameters]]</i>	Invokes an interactive shell on the local host. An optional command, with one or more optional parameters, can be given with the shell command.
\$ <i>Macro [Parameters]</i>	Executes the specified macro, previously defined with the macdef subcommand. Parameters are not expanded.
? <i>[Subcommand]</i>	Displays a help message describing the subcommand. If you do not specify a <i>Subcommand</i> parameter, the ftp command displays a list of known subcommands.
account <i>[Password]</i>	Sends a supplemental password that a remote host may require before granting access to its resources. If the password is not supplied with the command, the user is prompted for the password. The password is not displayed on the screen.
append <i>LocalFile [RemoteFile]</i>	Appends a local file to a file on the remote host. If the remote file name is not specified, the local file name is used, altered by any setting made with the ntrans subcommand or the nmap subcommand. The append subcommand uses the current values for form , mode , struct , and type subcommands while appending the file.
ascii	Synonym for the type ascii subcommand.
bell	Sounds a bell after the completion of each file transfer.
binary	Synonym for the type binary subcommand.
block	Synonym for the mode block subcommand.
bye	Ends the file-transfer session and exits the ftp command. Same as the quit subcommand.
carriage-control	Synonym for the form carriage-control subcommand.
case	Sets a toggle for the case of file names. When the case subcommand is On, the ftp command changes remote file names displayed in all capital letters from uppercase to lowercase when writing them in the local directory. The default is Off (so the ftp command writes uppercase remote file names in uppercase in the local directory).
cd <i>RemoteDirectory</i>	Changes the working directory on the remote host to the specified directory.
cdup	Changes the working directory on the remote host to the parent of the current directory.
close	Ends the file-transfer session, but does not exit the ftp command. Defined macros are erased. Same as the disconnect subcommand.
copylocal	Toggles local copy. copylocal defaults to off. An effort is made by ftp to make sure you do not zero out a file by ftp'ing it to itself (eg. same hostname, same pathname). Turning copylocal ON bypasses this check.
cr	Strips the carriage return character from a carriage return and line-feed sequence when receiving records during ASCII-type file transfers. (The ftp command terminates each ASCII-type record with a carriage return and line feed during file transfers.)

Records on remote hosts with operating systems other than the one you are running can have single line feeds embedded in records. To distinguish these embedded line feeds from record delimiters, set the **cr** subcommand to Off. The **cr** subcommand toggles between On and Off.

debug [0 1]	<p>Toggles debug record keeping On and Off. Specify debug or debug 1 to print each command sent to the remote host and save the restart control file. Specify debug again, or debug 0, to stop the debug record keeping. The Ctrl-C key sequence also saves the restart control file.</p> <p>Specifying the debug subcommand sends debugging information about ftp command operations to the syslogd daemon. If you specify the debug subcommand, you must edit the /etc/syslog.conf file and add one of the following entries:</p> <pre>user.info FileName</pre> <p>OR</p> <pre>user.debug FileName</pre> <p>Note: The syslogd daemon debug level includes info level messages.</p> <p>If you do not edit the /etc/syslog.conf file, no messages are produced. After changing the /etc/syslog.conf file, run the refresh -s syslogd or kill -1 SyslogdPID command to inform the syslogd daemon of the changes to its configuration file. For more information about debug levels, refer to the /etc/syslog.conf file. Also, refer to the ftp -d flag.</p>
delete <i>RemoteFile</i>	Deletes the specified remote file.
dir [<i>RemoteDirectory</i>] [<i>LocalFile</i>]	Writes a listing of the contents of the specified remote directory (<i>RemoteDirectory</i>) to the specified local file (<i>LocalFile</i>). If the <i>RemoteDirectory</i> parameter is not specified, the dir subcommand lists the contents of the current remote directory. If the <i>LocalFile</i> parameter is not specified or is a - (hyphen), the dir subcommand displays the listing on the local terminal.
disconnect	Ends the file-transfer session but does not exit the ftp command. Defined macros are erased. Same as the close subcommand.
ebcdic	Synonym for the type ebcdic subcommand.
exp_cmd	Toggles between conventional and experimental protocol commands. The default is off.
file	Synonym for the struct file subcommand.
form [carriage-control non-print telnet]	Specifies the form of the file transfer. The form subcommand modifies the type subcommand to send the file transfer in the indicated form. Valid arguments are carriage-control , non-print , and telnet .
	<p>carriage-control Sets the form of the file transfer to carriage-control.</p> <p>non-print Sets the form of the file transfer to non-print.</p> <p>telnet Sets the form of the file transfer to Telnet. Telnet is a Transmission Control Protocol/Internet Protocol (TCP/IP) protocol that opens connections to a system.</p>
get <i>RemoteFile</i> [<i>LocalFile</i>]	Copies the remote file to the local host. If the <i>LocalFile</i> parameter is not specified, the remote file name is used locally and is altered by any settings made by the case , ntrans , and nmap subcommands. The ftp command uses the current settings for the type , form , mode , and struct subcommands while transferring the file.

glob	<p>Toggles file-name expansion (globbing) for the mdelete, mget, and mput subcommands. If globbing is disabled, file-name parameters for these subcommands are not expanded. When globbing is enabled and a pattern-matching character is used in a subcommand that expects a single file name, results may be different than expected.</p> <p>For example, the append and put subcommands perform file-name expansion and then use only the first file name generated. Other ftp subcommands, such as cd, delete, get, mkdir, rename, and rmdir, do not perform file-name expansion and take the pattern-matching characters literally.</p> <p>Globbing for the mput subcommand is done locally in the same way as for the cs command. For the mdelete and mget subcommands, each file name is expanded separately at the remote machine and the lists are not merged. The expansion of a directory name can be different from the expansion of a file name, depending on the remote host and the ftp server.</p> <p>To preview the expansion of a directory name, use the mls subcommand: <pre>mls RemoteFile</pre></p>
hash	<p>To transfer an entire directory subtree of files, transfer a tar archive of the subtree in binary form, rather than using the mget or mput subcommand.</p> <p>Toggles hash sign (#) printing. When the hash subcommand is on, the ftp command displays one hash sign for each data block (1024 bytes) transferred.</p>
help [<i>Subcommand</i>]	Displays help information. See the ? subcommand.
image	Synonym for the type image subcommand.
lcd [<i>Directory</i>]	Changes the working directory on the local host. If you do not specify a directory, the ftp command uses your home directory.
local M	Synonym for the type local M subcommand.
ls [<i>RemoteDirectory</i>] [<i>LocalFile</i>]	Writes an abbreviated file listing of a remote directory to a local file. If the <i>RemoteDirectory</i> parameter is not specified, the ftp command lists the current remote directory. If the <i>LocalFile</i> parameter is not specified or is a - (hyphen), the ftp command displays the listing on the local terminal.
macdef <i>Macro</i>	<p>Defines a subcommand macro. Subsequent lines up to a null line (two consecutive line feeds) are saved as the text of the macro. Up to 16 macros, containing at most 4096 characters for all macros, can be defined. Macros remain defined until either redefined or a close subcommand is executed.</p> <p>The \$ (dollar sign) and \ (backslash) are special characters in ftp macros. A \$ symbol followed by one or more numbers is replaced by the corresponding macro parameter on the invocation line (see the \$ subcommand). A \$ symbol followed by the letter i indicates that the macro is to loop, with the \$i character combination being replaced by consecutive parameters on each pass.</p> <p>The first macro parameter is used on the first pass, the second parameter is used on the second pass, and so on. A \ symbol prevents special treatment of the next character. Use the \ symbol to turn off the special meanings of the \$ and \. (backslash period) symbols.</p>
mdelete <i>RemoteFiles</i>	Expands the files specified by the <i>RemoteFiles</i> parameter at the remote host and deletes the remote files.
mdir [<i>RemoteDirectories</i>] [<i>LocalFile</i>]	<p>Expands the directories specified by the <i>RemoteDirectories</i> parameter at the remote host and writes a listing of the contents of those directories to the file specified in the <i>LocalFile</i> parameter. If the <i>RemoteDirectories</i> parameter contains a pattern-matching character, the mdir subcommand prompts for a local file if none is specified. If the <i>RemoteDirectories</i> parameter is a list of remote directories separated by blanks, the last argument in the list must be either a local file name or a - (hyphen).</p> <p>If the <i>LocalFile</i> parameter is - (hyphen), the mdir subcommand displays the listing on the local terminal. If interactive prompting is on (see the prompt subcommand), the ftp command prompts the user to verify that the last parameter is a local file and not a remote directory.</p>

mget <i>RemoteFiles</i>	Expands the <i>RemoteFiles</i> parameter at the remote host and copies the indicated remote files to the current directory on the local host. See the glob subcommand for more information on file-name expansion. The remote file names are used locally and are altered by any settings made by the case , ntrans , and nmap subcommands. The ftp command uses the current settings for the form , mode , struct , and type subcommands while transferring the files.
mkdir [<i>RemoteDirectory</i>]	Creates the directory specified in the <i>RemoteDirectory</i> parameter on the remote host.
mls [<i>RemoteDirectories</i> <i>LocalFile</i>]	Expands the directories specified in the <i>RemoteDirectories</i> parameter at the remote host and writes an abbreviated file listing of the indicated remote directories to a local file. If the <i>RemoteDirectories</i> parameter contains a pattern-matching character, the mls subcommand prompts for a local file if none is specified. If the <i>RemoteDirectories</i> parameter is a list of remote directories separated by blanks, the last argument in the list must be either a local file name or a - (hyphen). If the <i>LocalFile</i> parameter is - (hyphen), the mls subcommand displays the listing on the local terminal. If interactive prompting is on (see the prompt subcommand), the ftp command prompts the user to verify that the last parameter is a local file and not a remote directory.
mode [stream block]	Sets file-transfer mode. If an argument is not supplied, the default is stream . block Sets the file-transfer mode to block. stream Sets the file-transfer mode to stream.
modtime	Shows the last modification time of the specified file on the remote machine. If the ftp command is not connected to a host prior to execution, the modtime subcommand terminates with an error message. The ftp command ignores parameter beyond the first parameter. If the <i>FileName</i> parameter is not specified, the ftp command prompts for a file name. If no file name is given, the ftp command sends a usage message to standard output and terminates the subcommand. If the name specified by the <i>FileName</i> parameter exists on the remote host, and the name specifies a file, then the ftp command sends a message containing the last modification time of the file to standard output and terminates the subcommand. If <i>FileName</i> specifies a directory, the ftp command sends an error message to standard output and terminates the subcommand. Note: The modtime subcommand interprets metacharacters when allowed.
mput [<i>LocalFiles</i>]	Expands the files specified in the <i>LocalFiles</i> parameter at the local host and copies the indicated local files to the remote host. See the glob subcommand for more information on file-name expansion. The local file names are used at the remote host and are altered by any settings made by the ntrans and nmap subcommands. The ftp command uses the current settings for the type , form , mode , and struct subcommands while transferring the files.
nlist [<i>RemoteDirectory</i>] [<i>LocalFile</i>]	Writes a listing of the contents of the specified remote directory (<i>RemoteDirectory</i>) to the specified local file (<i>LocalFile</i>). If the <i>RemoteDirectory</i> parameter is not specified, the nlist subcommand lists the contents of the current remote directory. If the <i>LocalFile</i> parameter is not specified or is a - (hyphen), the nlist subcommand displays the listing on the local terminal.

nmap [<i>InPattern</i> <i>OutPattern</i>]	<p>Turns the file-name mapping mechanism On or Off. If no parameters are specified, file-name mapping is turned off. If parameters are specified, source file names are mapped for the mget and mput subcommands and for the get and put subcommands when the destination file name is not specified. This subcommand is useful when the local and remote hosts use different file-naming conventions or practices. Mapping follows the pattern set by the <i>InPattern</i> and <i>OutPattern</i> parameters.</p> <p>The <i>InPattern</i> parameter specifies the template for incoming file names, which may have already been processed according to the case and ntrans settings. The template variables \$1 through \$9 can be included in the <i>InPattern</i> parameter. All characters in the <i>InPattern</i> parameter, other than the \$ (dollar sign) and the \ (backslash, dollar sign), are treated literally and are used as delimiters between <i>InPattern</i> variables. For example, if the <i>InPattern</i> parameter is \$1.\$2 and the remote file name is mydata.dat, the value of \$1 is mydata and the value of \$2 is dat.</p> <p>The <i>OutPattern</i> parameter determines the resulting file name. The variables \$1 through \$9 are replaced by their values as derived from the <i>InPattern</i> parameter, and the variable \$0 is replaced by the original file name. Additionally, the sequence [<i>Sequence1</i>,<i>Sequence2</i>] is replaced by the value of <i>Sequence1</i>, if <i>Sequence1</i> is not null; otherwise, it is replaced by the value of <i>Sequence2</i>. For example, the subcommand:</p> <pre>nmap \$1.\$2.\$3 [\$1,\$2].[\$2,file]</pre> <p>would yield myfile.data from myfile.data or myfile.data.old, myfile.file from myfile, and myfile.myfile from .myfile. Use the \ (backslash) symbol to prevent the special meanings of the \$ (dollar sign), [(left bracket),] (right bracket), and , (comma) in the <i>OutPattern</i> parameter.</p> <p>Synonym for the form non-print subcommand.</p>
non-print ntrans [<i>InCharacters</i> [<i>OutCharacters</i>]]	<p>Turns the file-name character translation mechanism On and Off. If no parameters are specified, character translation is turned off. If parameters are specified, characters in source file names are translated for mget and mput subcommands and for get and put subcommands when the destination file name is not specified.</p> <p>This subcommand is useful when the local and remote hosts use different file-naming conventions or practices. Character translation follows the pattern set by the <i>InCharacters</i> and <i>OutCharacters</i> parameter. Characters in a source file name matching characters in the <i>InCharacters</i> parameter are replaced by the corresponding characters in the <i>OutCharacters</i> parameter.</p> <p>If the string specified by the <i>InCharacters</i> parameter is longer than the string specified by the <i>OutCharacters</i> parameter, the characters in the <i>InCharacters</i> parameter are deleted if they have no corresponding character in the <i>OutCharacters</i> parameter.</p>
open <i>HostName</i> [<i>Port</i>]	<p>Establishes a connection to the FTP server at the host specified by the <i>HostName</i> parameter. If the optional port number is specified, the ftp command attempts to connect to a server at that port. If the automatic login feature is set (that is, the -n flag was not specified on the command line), the ftp command attempts to log in the user to the FTP server.</p> <p>You must also have a \$HOME/.netrc file with the correct information in it and the correct permissions set. The .netrc file must be in your home directory.</p>
passive	<p>Toggles passive mode for file transfers. When a file transfer command (such as get, mget, put, or mput) is invoked with passive mode off, the ftp server opens a data connection back to the client. In passive mode, the client opens data connections to the server when sending or receiving data.</p>
private	<p>Sets the protection level to "private." At this level, data is integrity and confidentially protected.</p>
prompt	<p>Toggles interactive prompting. If interactive prompting is on (the default), the ftp command prompts for verification before retrieving, sending, or deleting multiple files during the mget, mput, and mdelete subcommands. Otherwise, the ftp command acts accordingly on all files specified.</p>
protect	<p>This command returns the current level of protection.</p>

proxy [<i>Subcommand</i>]	Executes an ftp command on a secondary control connection. This subcommand allows the ftp command to connect simultaneously to two remote FTP servers for transferring files between the two servers. The first proxy subcommand should be an open subcommand to establish the secondary control connection. Enter the proxy ? subcommand to see the other ftp subcommands that are executable on the secondary connection.
	The following subcommands behave differently when prefaced by the proxy subcommand:
	<ul style="list-style-type: none"> • The open subcommand does not define new macros during the automatic login process. • The close subcommand does not erase existing macro definitions. • The get and mget subcommands transfer files from the host on the primary connection to the host on the secondary connection. • The put, mput, and append subcommands transfer files from the host on the secondary connection to the host on the primary connection. • The restart subcommand can be handled by the proxy command. • The status subcommand displays accurate information.
	File transfers require that the FTP server on the secondary connection must support the PASV (passive) instruction.
put <i>LocalFile</i> [<i>RemoteFile</i>]	Stores a local file on the remote host. If you do not specify the <i>RemoteFile</i> parameter, the ftp command uses the local file name to name the remote file, and the remote file name is altered by any settings made by the ntrans and nmap subcommands. The ftp command uses the current settings for the type , form , mode , and struct subcommands while transferring the files.
pwd	Displays the name of the current directory on the remote host.
quit	Closes the connection and exits the ftp command. Same as the bye subcommand.
quote <i>String</i>	Sends the string specified by the <i>String</i> parameter verbatim to the remote host. Execute the remotehelp or quote help subcommand to display a list of valid values for the <i>String</i> parameter. Note: "Quoting" commands that involve data transfers can produce unpredictable results.
record	Synonym for the struct record subcommand.
recv <i>RemoteFile</i> [<i>LocalFile</i>]	Copies the remote file to the local host. Same as the get subcommand.
reinitialize	Reinitializes an FTP session by flushing all I/O and allowing transfers to complete. Resets all defaults as if a user had just started an FTP session without logging in to a remote host.
remotehelp [<i>Subcommand</i>]	Requests help from the remote FTP server.
rename <i>FromName ToName</i>	Renames a file on the remote host.
reset	Clears the reply queue. This subcommand resynchronizes the command parsing.
restart get put append	Restarts a file transfer at the point where the last checkpoint was made. To run successfully, the subcommand must be the same as the aborted subcommand, including structure, type, and form. Valid arguments are get , put , and append .
rmdir <i>RemoteDirectory</i>	Removes the remote directory specified by the <i>RemoteDirectory</i> parameter at the remote host.
runique	(ReceiveUnique) Toggles the facility for creating unique file names for local destination files during get and mget subcommands. If this facility is Off (the default), the ftp command overwrites local files. Otherwise, if a local file has the same name as that specified for a local destination file, the ftp command modifies the specified name of the local destination file with .1 . If a local file is already using the new name, the ftp command appends the postfix .2 to the specified name. If a local file is already using this second name, the ftp command continues incrementing the postfix until it either finds a unique file name or reaches .99 without finding a unique file name. If the ftp command cannot find a unique file name, the ftp command reports an error and the transfer does not take place. Note that the runique subcommand does not affect local file names generated from a shell command.

safe	Sets the protection level to "safe." At this level, data is integrity protected.
send <i>LocalFile</i> [<i>RemoteFile</i>]	Stores a local file on the remote host. Same as the put subcommand.
sendport	Toggles the use of FTP PORT instructions. By default, the ftp command uses a PORT instruction when establishing a connection for each data transfer. When the use of PORT instructions is disabled, the ftp command does not use PORT instructions for data transfers. The PORT instruction is useful when dealing with FTP servers that ignore PORT instructions while incorrectly indicating the instructions have been accepted.
site <i>Args</i>	Displays or sets the idle time-out period, displays or sets the file-creation umask, or changes the permissions of a file, using the chmod command. Possible values for the <i>Args</i> parameter are umask and chmod .
size <i>RemoteFile</i>	Displays the size in bytes of the remote file specified by the <i>RemoteFile</i> parameter.
status	Displays the current status of the ftp command as well as the status of the subcommands.
stream	Synonym for the mode stream subcommand.
struct [file record]	Sets the data transfer structure type. Valid arguments are file and record . file Sets the data-transfer structure type to file. record Sets the data-transfer structure type to record.
sunique	(Send/Store Unique) Toggles the facility for creating unique file names for remote destination files during put and mput subcommands. If this facility is off (the default), the ftp command overwrites remote files. Otherwise, if a remote file has the same name as that specified for a remote destination file, the remote FTP server modifies the name of the remote destination file. Note that the remote server must support the STOU instruction.
system	Shows the type of operating system running on the remote machine.
telnet	Synonym for the form telnet subcommand.
tenex	Synonym for the type tenex subcommand.
trace	Toggles packet tracing.
type [ascii binary ebcdic image local M tenex]	Sets the file-transfer type. Valid arguments are ascii , binary , ebcdic , image , local M , and tenex . If an argument is not specified, the current type is printed. The default type is ascii ; the binary type can be more efficient than ascii . ascii Sets the file-transfer type to network ASCII. This type is the default. File transfer may be more efficient with binary-image transfer. See the binary argument for further information. binary Sets the file-transfer type to binary image. This type can be more efficient than an ASCII transfer. ebcdic Sets the file-transfer type to EBCDIC. image Sets the file-transfer type to binary image. This type can be more efficient than an ASCII transfer. local M Sets the file-transfer type to local. The <i>M</i> parameter defines the decimal number of bits per machine word. This parameter does not have a default. tenex Sets the file-transfer type to that needed for TENEX machines.
user <i>User</i> [<i>Password</i>] [<i>Account</i>]	Identifies the local user (<i>User</i>) to the remote FTP server. If the <i>Password</i> or <i>Account</i> parameter is not specified and the remote server requires it, the ftp command prompts for the password or account locally. If the <i>Account</i> parameter is required, the ftp command sends it to the remote server after the remote login process completes. Note: Unless automatic login is disabled by specifying the -n flag on the command line, the ftp command sends the <i>User</i> , <i>Password</i> , and <i>Account</i> parameters automatically for the initial connection to the remote server. You also need a .netrc file in your home directory in order to issue an automatic login.

verbose

Toggles verbose mode. When the verbose mode is on (the default), the **ftp** command displays all responses from the remote FTP server. Additionally, the **ftp** command displays statistics on all file transfers when the transfers complete.

Examples

1. To invoke the **ftp** command, log in to the system **canopus**, display local help information, display remote help information, display status, toggle the **bell**, **prompt**, **runique**, **trace**, and **verbose** subcommands, and then quit, enter:

```
$ ftp canopus
Connected to canopus.austin.century.com.
220 canopus.austin.century.com FTP server
(Version 4.1 Sat Nov 23 12:52:09 CST 1991) ready.
Name (canopus:eric): dee
331 Password required for dee.
Password:
230 User dee logged in.
ftp> help
Commands may be abbreviated. Commands are:
!      delete      mdelete      proxy        runique
$      debug        mdir         sendport     send
account dir          mget         put          size
append disconnect  mkdir        pwd          status
ascii  form          mls          quit         struct
bell   get           mode         quote        sunique
binary glob          modtime     recv         system
bye    hash          mput        remotehelp   tenex
case   help          nmap        rstatus      trace
cd     image         nlist       rhelp        type
cdup   lcd           ntrans      rename       user
close  ls            open        reset        verbose
cr     macdef        prompt      rmdir        ?
clear  private       protect     safe

ftp> remotehelp
214-The following commands are recognized(* =>'s unimplemented).
USER PORT RETR MSND* ALLO DELE SITE* XMKD CDUP
PASS PASV STOR MSOM* REST* CWD STAT* RMD XCUP
ACCT* TYPE APPE MSAM* RNFR XCWD HELP XRMD STOU
REIN* STRU MLFL* MRSQ* RNTD LIST NOOP PWD
QUIT MODE MAIL* MRCP* ABOR NLST MKD XPWD
AUTH ADAT PROT PBSZ MIC ENC CCC

214 Direct comments to ftp-bugs@canopus.austin.century.com.
ftp> status
Connected to canopus.austin.century.com.
No proxy connection.
Mode: stream; Type: ascii; Form: non-print; Structure: file
Verbose: on; Bell: off; Prompting: on; Globbing: on
Store unique: off; Receive unique: off
Case: off; CR stripping: on
Ntrans: off
Nmap: off
Hash mark printing: off; Use of PORT cmds: on
ftp> bell
Bell mode on.
ftp> prompt
Interactive mode off.
ftp> runique
Receive unique on.
ftp> trace
Packet tracing on.
ftp> verbose
Verbose mode off.
ftp> quit
$
```

- To invoke the **ftp** command, log in to the system canopus, print the working directory, change the working directory, set the file transfer type to ASCII, send a local file to the remote host, change the working directory to the parent directory, and then quit, enter:

```
$ ftp canopus
Connected to canopus.austin.century.com.
220 canopus.austin.century.com FTP server
(Version 4.1 Sat Nov 23 12:52:09 CST 1991) ready.
Name (canopus:eric): dee
331 Password required for dee.
Password:
230 User dee logged in.
ftp> pwd
257 "/home/dee" is current directory.
ftp> cd desktop
250 CWD command successful.
ftp> type ascii
200 Type set to A.
ftp> send typescript
200 PORT command successful.
150 Opening data connection for typescript (128.114.4.99,1412).
226 Transfer complete.
ftp> cdup
250 CWD command successful.
ftp> bye
221 Goodbye.
$
```

- To invoke the **ftp** command with automatic logon (using the **.netrc** file), open a session with the system canopus, log in, change the working directory to the parent directory, print the working directory, list the contents of the current directory, delete a file, write a listing of the contents of the current directory to a local file, close the session, and then quit, enter:

```
$ ftp canopus
Connected to canopus.austin.century.com.
220 canopus.austin.century.com FTP server
(Version 4.1 Sat Nov 23 12:52:09 CST 1991) ready.
331 Password required for dee.
230 User dee logged in.
ftp> cdup
250 CWD command successful.
ftp> pwd
257 "/home" is current directory.
ftp> dir
200 PORT command successful.
150 Opening data connection for /usr/bin/ls (128.114.4.99,1407)
(0 bytes).
total 104
drwxr-xr-x  2 system      32 Feb 23 17:55 bin
Drwxr-xr-x 26 rios       4000 May 30 17:18 bin1
drwxr-xr-x  2 system      32 Feb 23 17:55 books
drwxrwxrwx 18 rios       1152 Jun  5 13:41 dee
-r--r--r--  1 system     9452 May 17 12:21 filesystems
drwxr-xr-x  2 system      32 Feb 23 17:55 jim
drwxr-xr-x  5 system       80 Feb 23 17:55 krs
drwxrwxrwx  2 rios       16432 Feb 23 17:36 lost+found
-rwxr-xr-x  1 rios       3651 May 24 16:45 oldmail
drwxr-xr-x  2 system     256 Feb 23 17:55 pubserv
drwxrwxrwx  2 system     144 Feb 23 17:55 rein989
drwxr-xr-x  2 system     112 Feb 23 17:55 reinstall
226 Transfer complete.
ftp> delete oldmail
250 DELE command successful.
ftp> mdir /home/dee/bin binlist
output to local-file: binlist? y
200 PORT command successful.
150 Opening data connection for /usr/bin/ls (128.114.4.99,1408) (0 bytes).
226 Transfer complete.
```

```
ftp> close
221 Goodbye.
ftp> quit
$
```

Files

<code>/usr/samples/tcpip/netrc</code>	Contains the sample <code>.netrc</code> file.
<code>/etc/syslog.conf</code>	Contains configuration information for the <code>syslogd</code> daemon.

Related Information

The `stty` command.

grep command

Purpose

Searches for a pattern in a file.

Syntax

```
grep [ -E | -F ] [ -i ] [ -h ] [ -H ] [ -L ] [ -r | -R ] [ -s ] [ -v ] [ -w ] [ -x ] [ -y ] [ [ [ -b ] [ -n ] ] | [ -c | -l | -q ] ] [ -p [ Separator ] ] { [ -e PatternList ... ] [ -f PatternFile ... ] | PatternList ... } [ File ... ]
```

Description

The `grep` command searches for the pattern specified by the *Pattern* parameter and writes each matching line to standard output. The patterns are limited regular expressions in the style of the `ed` or `egrep` command. The `grep` command uses a compact non-deterministic algorithm.

The `grep` command displays the name of the file containing the matched line if you specify more than one name in the *File* parameter. Characters with special meaning to the shell (`$`, `*`, `[`, `|`, `^`, `(`, `)`, `\`) must be in quotation marks when they appear in the *Pattern* parameter. When the *Pattern* parameter is not a simple string, you usually must enclose the entire pattern in single quotation marks. In an expression such as `[a-z]`, the `-` (minus sign) *cml* specifies a range, according to the current collating sequence. A collating sequence may define equivalence classes for use in character ranges. If no files are specified, `grep` assumes standard input.

Notes:

1. Do not run the `grep` command on a special file because it produces unpredictable results.
2. Input lines should not contain the NULL character.
3. Input files should end with the newline character.
4. The newline character will not be matched by the regular expressions.
5. Although some flags can be specified simultaneously, some flags override others. For example, the `-l` option takes precedence over all other flags. And if you specify both the `-E` and `-F` flags, the last one specified takes priority.

Flags

<code>-b</code>	Precedes each line by the block number on which it was found. Use this flag to help find disk block numbers by context. The <code>-b</code> flag cannot be used with input from stdin or pipes.
<code>-c</code>	Displays only a count of matching lines.

-E	Treats each pattern specified as an extended regular expression (ERE). A NULL value for the ERE matches every line.
-e <i>PatternList</i>	Specifies one or more search patterns. This works like a simple pattern but is useful when the pattern begins with a - (minus). Patterns should be separated by a new-line character. A NULL pattern can be specified by two adjacent new-line characters or a quotation mark followed by a new-line character (" <code>\n</code> "). Each pattern is treated like a basic regular expression (BRE) unless the -E or -F flag is also specified. Multiple -e and -f flags are accepted by grep . All of the specified patterns are used when matching lines, but the order of evaluation is unspecified.
-F	Treats each specified pattern as a string instead of a regular expression. A NULL string matches every line.
-f <i>PatternFile</i>	Specifies a file containing search patterns. Each pattern should be separated by a new-line character, and an empty line is considered a NULL pattern. Each pattern is treated like a basic regular expression (BRE), unless the -E or -F flag is also specified.
-h	Prevents the name of the file containing the matching line from being appended to that line. Suppresses file names when multiple files are specified.
-H	If the -r or -R option is specified and a symbolic link referencing a file of type directory is specified on the command line, grep will search the files of the directory referenced by the symbolic link and all the files in the file hierarchy below it.
-i	Ignores the case (uppercase or lowercase) of letters when making comparisons.
-l	Lists just the names of files (once) which contain matching lines. Each file name is separated by a new-line character. If standard input is searched, a path name of (StandardInput) is returned. The -l flag with any combination of the -c and -n flags behaves like the -l flag only.
-L	If the -r or -R option is specified and a symbolic link referencing a file of type directory is specified on the command line or encountered during the traversal of a file hierarchy, grep shall search the files of the directory referenced by the symbolic link and all the files in the file hierarchy below it. If both -H and -L are specified, the last option specified on the command line takes effect.
-n	Precedes each line with the relative line number in the file. Each file starts at line 1, and the line counter is reset for each file processed.
-p [<i>Separator</i>]	Displays the entire paragraph containing matched lines. Paragraphs are delimited by paragraph separators, as specified by the <i>Separator</i> parameter, which are patterns in the same form as the search pattern. Lines containing the paragraph separators are used only as separators; they are never included in the output. The default paragraph separator is a blank line.
-q	Suppresses all writing to standard output, regardless of matching lines. Exits with a zero status if an input line is selected. The -q flag with any combination of the -c , -l and -n flags behaves like the -q flag only.
-r	Searches directories recursively. By default, links to directories are followed.
-R	Searches directories recursively. By default, links to directories are not followed.
-s	Suppresses error messages ordinarily written for nonexistent or unreadable files. Other error messages are not suppressed.
-v	Displays all lines not matching the specified pattern.
-w	Does a word search.
-x	Displays lines that match the specified pattern exactly with no additional characters.
-y	Ignores the case of letters when making comparisons.
<i>PatternList</i>	Specifies one or more patterns to be used during the search. The patterns are treated as if they were specified using the -e flag.
<i>File</i>	Specifies a name of a file to be searched for patterns. If no <i>File</i> variable is given, the standard input is used.

Exit Status

This command returns the following exit values:

- 0 A match was found.
- 1 No match was found.
- >1 A syntax error was found or a file was inaccessible (even if matches were found).

Examples

1. To use a pattern that contains some of the pattern-matching characters *, ^, ?, [,], \(\, \), \{, and \}, enter:

```
grep "[a-zA-Z]" pgm.s
```

This displays every line in `pgm.s` whose first character is a letter.

2. To display all lines that do not match a pattern, enter:

```
grep -v "^#" pgm.s
```

This displays every line in `pgm.s` whose first character is not a # (pound sign).

3. To display all lines in the `file1` file that match either the `abc` or `xyz` string, enter:

```
grep -E "abc|xyz" file1
```

4. To search for a \$ (dollar sign) in the file named `test2`, enter:

```
grep \$ test2
```

The `\$` (double backslash) characters are necessary in order to force the shell to pass a `\$` (single backslash, dollar sign) to the **grep** command. The `\` (single backslash) character tells the **grep** command to treat the following character (in this example the `$`) as a literal character rather than an expression character. Use the **fgrep** command to avoid the necessity of using escape characters such as the backslash.

5. To search recursively through `/tmp` to find files which have the word `IBM` without recursing through links pointing to directories, type:

```
grep -R IBM /tmp
```

OR

```
grep -r -H IBM /tmp
```

6. To search recursively through `/tmp` to find files which have the word `IBM` and recurse through links as well, type:

```
grep -r IBM /tmp
```

OR

```
grep -R -L IBM /tmp
```

Files

`/usr/bin/grep` Contains the **grep** command.

head command

Purpose

Displays the first few lines of a file.

Syntax

```
head [ -Count | -cNumber | -n Number ] [ File ... ]
```

Description

The **head** command writes to standard output a specified number of lines or bytes of each of the specified files, or of the standard input. If no flag is specified with the **head** command, the first 10 lines are displayed by default. The *File* parameter specifies the names of the input files. An input file must be a text file. When more than one file is specified, the start of each file will look like the following:

```
==> filename <==
```

To display a set of short files, identifying each one, enter:

```
example% head -9999 filename1 filename2...
```

Flags

<i>-Count</i>	Specifies the number of lines from the beginning of each specified file to be displayed. The <i>Count</i> variable must be a positive decimal integer. This flag is equivalent to the <i>-n Number</i> flag, but should not be used if portability is a consideration.
<i>-c Number</i>	Specifies the number of bytes to display. The <i>Number</i> variable must be a positive decimal integer.
<i>-n Number</i>	Specifies the number of lines from the beginning of each specified file to be displayed. The <i>number</i> variable must be a positive decimal integer. This flag is equivalent to the <i>-Count</i> flag.

Exit Status

This command returns the following exit values:

0	Successful completion.
>0	An error occurred.

Examples

To display the first five lines of the Test file, enter:

```
head -5 Test
```

OR

```
head -n 5 Test
```

Related Information

The **tail** command.

hostmap command

Purpose

Directly manipulates address-mapping entries in the system configuration database.

Syntax

To Add an Address-to-Host Name Mapping

```
hostmap -addr IPAddress -host HostName...
```

To Delete an Address-to-Host Name Mapping

```
hostmap -rm IPAddress
```

To Show all Address-to-Host Name Mappings

```
hostmap -ls
```

Description

The **hostmap** low-level command adds, deletes, or lists address-mapping entries in the system configuration database. Entries in the database are used to map an Internet Protocol (IP) address (local or remote) to its equivalent host names.

An Internet Protocol (IP) address of a given local or remote host may be associated with one or more host names. Represent an IP address in dotted decimal format. Represent a host name as a string with a maximum length of 255 characters, and do not use any blank characters.

Note:

1. Valid host names or alias host names must contain at least one alphabetic character. If you choose to specify a host name or alias that begins with an x followed by any hexadecimal digit (0-f), the host name or alias must also contain at least one additional letter that cannot be expressed as a hexadecimal digit. The system interprets a leading x followed by a hexadecimal digit as the base 16 representation of an address unless there is at least one character in the host name or alias that is not a hexadecimal digit. Thus, xdeer would be a valid host name, whereas xdee would not.
2. The **hostmap** command does not recognize the following addresses: .08, .008, .09, and .009. Addresses with leading zeros are interpreted as octal, and numerals in octal cannot contain 8s or 9s.

Flags

-addr IPAddress	Adds an IP address-to-host name mapping entry for the given Internet Protocol address in the database. Specify the host names with the -host flag.
-host HostName...	Specifies a list of host names. Entries in the list should be separated by blanks.
-ls	Shows all entries in the database.
-rm IPAddress	Deletes the IP address-to-host name mapping entry in the database that corresponds to the given address specified by the IPAddress variable.

Exit Status

See "Exit status for Virtual I/O Server commands" on page 1.

Examples

1. To add an entry in the database associating an address with a series of host names, enter the command in the following format:
hostmap -addr 192.100.201.7 -host alpha bravo charlie

The IP address 192.100.201.7 is specified as the address of the host that has a primary host name of alpha with synonyms of **bravo** and **charlie**.

Note: If you attempt to use .08, .008, .09, or .009 in an address to add, you will get an error message that states "IP Address already exists," although the address is not in the database.

2. To list all entries in the database, enter the command in the following format:

```
hostmap -ls
```

Related Information

The hostname command, and the mktpip command.

hostname command

Purpose

Sets or displays the name of the current host system.

Syntax

```
hostname [ HostName ]
```

Description

The **hostname** command sets or displays the name of the current host system.

Parameters

HostName Sets the primary name of the host.

Exit Status

See "Exit status for Virtual I/O Server commands" on page 1.

Examples

1. To set the hostname to **rotterdam**, type:

```
hostname rotterdam
```

Related Information

The **mktpip** command, the **startnet** command, the **stopnet** command, the **cfglnagg** command, the **netstat** command, the **entstat** command, the **cfgnamesrv** command, the **hostmap** command, the **traceroute** command, the **ping** command, the **optimizenet** command.

importvg command

Purpose

Imports a new volume group definition from a set of physical volumes.

Syntax

```
importvg [ -vg VolumeGroup ] PhysicalVolume
```

Description

The **importvg** command makes the previously exported volume group known to the system. The *PhysicalVolume* parameter specifies only one physical volume to identify the volume group; any remaining physical volumes (those belonging to the same volume group) are found by the **importvg** command and included in the import. An imported volume group is automatically activated. When a volume group with file systems is imported, the */etc/filesystems* file is updated with values for the new logical volumes and mount points.

After importing the volume group, you must run the **fsck** command before the file systems can be mounted. Care should be taken to avoid using mount point longer than 128 characters as the mount point information would be missing from the LVCB (logical volume control block) if it is longer than 128 characters. In this case, the **importvg** command will not be able to update the */etc/filesystems* file with the stanza for the newly imported logical volume.

The **importvg** command changes the name of a logical volume if the name already exists in the system. It prints a message and the new name to standard error, and updates the */etc/filesystems* file to include the new logical volume name.

Flags

-vg *VolumeGroup* Specifies the name to use for the new volume group. If this flag is not used, the system automatically generates a new name.

The volume group name can only contain the following characters: "A" through "Z," "a" through "z," "0" through "9," or "_" (the underscore), "-" (the minus sign), or "." (the period). All other characters are considered invalid.

Exit Status

See "Exit status for Virtual I/O Server commands" on page 1.

Examples

1. To import the volume group **bkvg** from physical volume **hdisk07**, type:

```
importvg -vg bkvg hdisk07
```

The volume group **bkvg** is made known to the system.

Restrictions

Mount points cannot be longer than 128 characters.

Related Information

The **activatevg** command, the **chvg** commands, the **deactivatevg** command, the **exportvg** command, the **extendvg** command, the **lsvg** command, the **mirrorios** command, the **mkvg** command, the **syncvg** command, and the **unmirrorios** command.

installios command

Purpose

Installs the Virtual I/O Server. This command is run from the HMC.

Syntax

`installios [-p partition_name -i ipaddr or hostname -S subnet_mask -g gateway -d path -s system_name -m mac_address -r profile [-n] [-P speed] [-D duplex] [-l language]] [-t] | -u`

Description

The `installios` command installs the Virtual I/O Server. It must be run from the HMC. All of the flags are optional. If no flags are provided, the `installios` wizard will be invoked and the user will be prompted to interactively enter the information contained in the flags.

Flags

- s** Specifies the managed system. The name of the managed system maintained by the HMC. This name must match the name shown on the HMC, not a host name.
- p** Specifies the partition name. The name of the LPAR that will be installed with Virtual I/O Server. This partition must be of type Virtual I/O Server and the name given for it must match the name shown on the HMC, not a host name.
- r** Specifies the profile name. The name of the profile that contains the hardware resources being installed to.
- d** Specifies the path to installation images. Either `/dev/cdrom` or the path to a system backup of the Virtual I/O Server created by the `backupios` command. The path may also specify a remote location mountable by NFS such as `hostname:path_to_backup`
- i** Specifies the client IP address. The IP address with which the client's network interface will be configured for network installation of the Virtual I/O Server operating system.
- S** Specifies the client subnet mask. The subnet mask with which the client's network interface will be configured for network installation of the Virtual I/O Server operating system.
- g** Specifies the client gateway. The default gateway that the client will use during network installation of the Virtual I/O Server operating system.
- m** Specifies the client MAC address. The MAC address of the client network interface through which the network installation of the Virtual I/O Server will take place.
- P** Specifies speed (optional) The communication speed with which to configure the client's network interface. This value can be 10, 100, or 1000, and is 100 by default if this flag is not specified.
- D** Specifies duplex (optional). The duplex setting with which to configure the client's network interface. This value can be `full` or `half` and is set to `full` by default if this flag is not specified.
- n** Specifies not to configure the client's network interface (optional): If this flag is specified, then the client's network interface will not be configured with the IP settings specified in the flags given to this command after the installation has completed.
- l** Specifies language (optional): The language in which the license agreement will be displayed before the installation. Upon viewing the license, a prompt will be shown asking if the license is to be accepted. If the prompt is answered with `y`, then the installation will proceed and the Virtual I/O Server license will be automatically accepted after the installation. If the prompt is answered with `n`, then the `installios` command will exit and the installation will not proceed. If this flag is not specified, then the installation will proceed, but the Virtual I/O Server will not be usable until the license is manually accepted after the installation.
- t** Specifies to migrate the Virtual I/O Server. This option creates network installation management (NIM) resources to be used for the migration and requires a Virtual I/O Server migration DVD.
- u** Unconfigure `installios` (optional). Will manually unconfigure the `installios` installation resources. This flag is only needed if a problem occurs during the installation and `installios` does not automatically unconfigure itself.

invscout command

Purpose

Surveys the host system for currently installed microcode or Vital Product Data (VPD).

Syntax

```
invscout [-vpd | -report] [-model Type-Model] [-serial SerialNumber]
```

```
invscout -version
```

```
invscout -rpm rpmPackage rpmOption ...
```

```
invscout -install Device [-file FileName]
```

Description

The **invscout** command executes one instance of the stand alone version of the Inventory Scout process. The **invscout** command starts the server daemon side of a client-server version. The Inventory scout process supports two survey types:

- Microcode Survey
- Vital Product Data (VPD) Survey

Flags

-file <i>FileName</i>	Specifies the location of the upgrade.
-install <i>Device</i>	Installs microcode for a given device.
-model <i>Type-Model</i>	Machine type and model. For a VPD survey, allows input of the host platform machine type and model for hosts that use this information.
-report	For a Microcode Survey, sends a copy of the formatted text report file to the screen from which the command was invoked. This flag is ignored if the -vpd flag is used.
-rpm <i>rpmPackage</i>	RPM options: <ul style="list-style-type: none">-i Installs a new package.-e Uninstalls a package.-qp Query for an uninstalled package.--force Used to install a package even if it is already installed.--ignoreos Allows installation of a package even if the operating system of the host and binary RPM are different.
-serial <i>SerialNumber</i>	Serial number. For a VPD survey, allows input of the host serial number for hosts that use this information.
-version	Displays the versions of this command and of the logic database currently in use.
-vpd	Sets the survey or concatenation type to VPD (the default is Microcode).

Exit Status

The following exit values are returned:

0 Successful completion.

>0

An error occurred.

Examples

1. To generate report on microcode levels of all the devices, type:
`invscout -report`
2. To get the vpd survey of the partition. type:
`invscout -vpd`

ioslevel Command

Purpose

Reports the latest installed maintenance level of the system.

Syntax

`ioslevel`

Description

The `ioslevel` command displays the Virtual I/O Server level.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1

Related Information

The `lssw` command, the `updateios` command, the `remote_management` command, the `oem_setup_env` command, and the `oem_platform_level` command.

ldapadd command

Purpose

Opens a connection to an LDAP server, binds, and modifies or adds entries.

Syntax

`ldapadd [-a] [-b] [-c] [-C] [-d] [-D] [-f] [-g] [-G] [-h] [-i file] [-K] [-m] [-M] [-N] [-O] [-p] [-P] [-r] [-R] [-U] [-v] [-V] [-w] [-y] [-Y] [-Z]`

Description

The `ldapmodify` command opens a connection to an LDAP server, binds, and modifies or adds entries. The entry information is read from standard input or from file, specified using the `-f` option. `ldapadd` is implemented as a hard link to the `ldapmodify` tool. When invoked as `ldapadd`, the `-a` (add new entry) option is turned on automatically.

Flags

- | | |
|-----------------|--|
| <code>-a</code> | Forces add operation as default. |
| <code>-b</code> | Supports binary values from files (old style paths). |
| <code>-c</code> | Specifies continuous operation; do not stop processing on error. |

-C <i>charset</i>	Sets the character set name to use, as registered with Internet Assigned Numbers Authority (IANA).
-d <i>level</i>	Sets the debugging level in LDAP library.
-D <i>dn</i>	Binds dn.
-f <i>file</i>	Specifies that entry-modification information should be read from the specified file. Note: Standard input is used if the file is not specified.
-g	Specifies that trailing spaces should not be stripped on attribute values.
-G <i>realm</i>	Specifies that realm is to be used for the DIGEST-MD5 bind mechanism.
-h <i>host</i>	Specifies the LDAP server host name.
-i <i>file</i>	Specifies that read entry-modification information should be read from the specified file. Note: Standard input is used if the file is not specified.
-K <i>keyfile</i>	Specifies the file to use for keys.
-m <i>mechanism</i>	Performs SASL bind with the given mechanism.
-M	Manages referral objects as normal entries.
-N <i>key_name</i>	Specifies the private key name to use in the keyfile.
-O <i>maxhops</i>	Specifies the maximum number of referrals to follow in a sequence.
-p <i>port</i>	Specifies the LDAP server port number.
-P <i>key_pw</i>	Specifies the keyfile password.
-r	Forces replace operation as default.
-R	Specifies not to chase referrals.
-U <i>username</i>	Specifies the user name for the DIGEST-MD5 bind mechanism.
-v	Specifies verbose mode
-V <i>version</i>	Specifies the LDAP protocol version (2 or 3; default is 3).
-w <i>password</i>	Binds password or '?' for non-echoed prompt.
-y	Requests proxied authorization control.
-Y	Uses a secure ldap connection (TLS).
-Z	Uses a secure ldap connection (SSL).

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Related Information

The **mkldap** command, and the **ldapsearch** command.

ldapsearch command

Purpose

Opens a connection to an LDAP server, binds, and performs a search using the filter *filter*.

Syntax

```
ldapsearch [-a ][-A ][-b basedn][-B][-C][-d][-D ][-e ][-f][-F][-G][-h][-i][-k][-K][-l][-L][-m][-M][-n][-N][-o][-O][[-p][[-P][[-q][[-R][[-s][[-t][[-T][[-U][[-v][[-V][[-w][[-y][[-Y][[-z ][[-Z][[-9] filter [attributes...]
```

where *basedn* equals the base distinguished name for the search, *filter* equals the LDAP search filter, options equals any other flags, and *attributes* equals a whitespace-separated list of attributes to retrieve. If no attribute list is specified, all attributes are retrieved.

Note: *basedn* is optional if LDAP_BASEDN is set in the environment.

Description

If the `ldapsearch` command finds one or more entries, the attributes specified by *attrs* are retrieved and the entries and values are printed to standard output. If no *attrs* are listed, all attributes are returned.

Flags

-a <i>deref</i>	Specifies how to dereference aliases (never, always, search, or find).
-A	Retrieves attribute names only (no values).
-b <i>basedn</i>	Specifies base distinguished name (dn) for search. LDAP_BASEDN in environment is the default.
-B	Specifies not to suppress printing of non-ASCII values.
-C <i>charset</i>	Sets the character set name to use, as registered with Internet Assigned Numbers Authority (IANA).
-d <i>level</i>	Sets the debugging level in LDAP library.
-D <i>dn</i>	Binds dn.
-e	Displays LDAP library version information and quit.
-f <i>file</i>	Performs sequence of searches using filters in 'file' " must be substituted for the filter.
-F <i>sep</i>	Prints 'sep' between attribute names and values.
-G <i>realm</i>	Specifies that realm is to be used for the DIGEST-MD5 bind mechanism.
-h <i>host</i>	Specifies the LDAP server host name.
-i <i>file</i>	Performs sequence of searches using filters in 'file' " must be substituted for the filter.
-k	Uses server administration control on bind.
-K <i>keyfile</i>	Specifies the file to use for the keys.
-l <i>time</i>	Specifies time limit (in seconds) for search.
-L	Prints entries in LDIF format (-B is implied).
-m <i>mechanism</i>	Performs SASL bind with the specified mechanism.
-M	Manages referral objects as normal entries.
-n	Shows what would be done but don't actually perform the action.
-N <i>key_name</i>	Specifies the private key name to use in the keyfile.
-o <i>attr_type</i>	Sorts based on specified attribute type.
-O <i>maxhops</i>	Specifies the maximum number of referrals to follow in a sequence.
-p <i>port</i>	Specifies the LDAP server port number.
-P <i>key_pw</i>	Specifies the keyfile password.
-q <i>pagesize</i>	Queries page size for paged results.
-R	Specifies not to chase referrals.
-s <i>scope</i>	Searches scope. The search scope can be one of the following: base, one, or sub.
-t <i>scope</i>	Writes values to files in /tmp.
-T <i>seconds</i>	Specifies the number of seconds to wait between pages for paged results.
-U <i>username</i>	Specifies the user name for the DIGEST-MD5 bind mechanism.
-v	Runs in verbose mode.
-V <i>version</i>	Specifies the LDAP protocol version. The version can be 2 or 3. The default is 3.
-w <i>password</i>	Binds password or '?' for non-echoed prompt.
-y <i>proxydn</i>	Sets proxied ID for proxied authorization operation.
-Y	Uses a secure LDAP connection (TLS).
-zsize	Specifies the size limit (in entries) for the search.
-Z	Uses a secure LDAP connection (SSL).
-9control	Sets the criticality for control option. The control can be one of the following options: <ul style="list-style-type: none">• s sets criticality for sorting to false• p sets criticality for paging to false

Exit Status

See "Exit status for Virtual I/O Server commands" on page 1.

Examples

To open a connection to the LDAP server and search on specific attributes, type:
`ldapsearch -h vclient.host.com -D cn=admin -w adminpw -b cn=aixdata objectclass=*`

Related Information

The `mkldap` command and the `ldapadd` command.

ldfware command

Purpose

Loads the system's flash EPROM with the specified file, which must contain a valid binary flash EPROM image, and then reboots the system.

Syntax

```
ldfware [ -dev Device ] -file filename
```

`ldfware -commit`

`ldfware -reject`

Description

Periodically, you need to install fixes for your server firmware. If you do not use an HMC to manage your server, you must get your fixes through your operating system. If your server is not connected to the Internet, you must obtain optical media that contains the server firmware fix and install the server firmware fix from the optical media. The flash update image file is copied to the file system from diskette. The user needs to provide the image on a diskette if the user does not have access to remote file systems or any other files that are on the system. If enough space is not available, an error is reported stating additional system memory is needed. After the file is copied, a warning screen asks for confirmation to continue the update flash. Continuing the update flash reboots the system. The current flash image is not saved.

After you download and install a firmware fix, the fix is temporarily installed until you install it permanently. You might want to use the new level of firmware for a period of time to verify that it works correctly. When you are sure that the new level of firmware works correctly, you can permanently install the firmware fix. Be aware that if you install the firmware fix permanently (copy the temporary firmware level from the temporary side to the permanent side, so that the temporary and permanent sides contain the same level of firmware), you cannot return to the level that was previously on the permanent side. To install it permanently, use the **-commit** flag.

Attention: The `ldfware` command reboots the entire system. Do not use this command if more than one user is signed onto the system.

Flags

-commit	Commits the temporary image when booted from the temporary image. This effectively causes the permanent image to be copied over by the temporary image. This flag is not supported on all systems.
-dev <i>Device</i>	Specifies that the flash update image file is on diskette. The <code>device_name</code> variable specifies the device. The default <code>device_name</code> is <code>/dev/fd0</code> .

-file <i>FileName</i>	Specifies the file name of the image file source. The flash update image file source follows this flag. The <code>file_name</code> variable specifies the fully qualified path of the flash update image file.
-reject	Rejects the temporary image when booted from the permanent image. This overwrites the temporary image with the permanent image. This flag is not supported on all systems.

Exit Status

The following exit codes are returned:

0	Completed successfully.
1	Command terminated due to an error.
2	Command was aborted by the user.
3	Command was aborted by the user using the F10 key.

Examples

- To update the firmware from a local file, type the following command:

```
ldfware -file /tmp/firmware/latest_flash
```
- To update the firmware that is on a diskette, type the following command:

```
ldfware -device /dev/fd0 -file latest_flash
```
- To update the firmware that is on a CD device, type the following command:

```
mount -cd /mnt ldfware -file /mnt/firmware/latest_flash
```

license command

Purpose

View and accept the license agreement.

Syntax

```
license { [ -view ] [ -accept ] } [ -lang Name ]
```

```
license [ -ls ]
```

Description

The **license** command is used to view and accept the Virtual I/O Server license agreement. If no flags are specified, the current status of the license agreement is displayed. If the license has been accepted, the date and time of acceptance is displayed.

Flags

-accept	Accepts the license agreement.
-lang <i>Name</i>	Specifies the language-territory (locale name) that the license will be displayed in. The default is <code>en_US</code> .
-ls	Lists available languages.
-view	Displays the Virtual I/O Server license agreement.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

1. To view the license in the en_US locale, type:
license -view
2. To accept the license in the fr_FR locale, type:
license -accept -lang fr_FR
3. To view if the license has been accepted, type:
license

Related Information

The chlang command.

loadopt command

Purpose

Load a virtual optical media disk from the Virtual Media Repository into a virtual optical device.

Syntax

```
loadopt [ -f ] [ -release ] -disk FileName -vtd VirtualTargetDevice
```

Description

The **loadopt** command loads the specified virtual optical disk (*FileName*) into the specified virtual optical device (*VirtualTargetDevice*).

Note: A particular virtual disk can only be loaded into more than one file-backed virtual optical device simultaneously if the virtual DVD is designated as read-only. If the virtual optical device already has a different virtual disk loaded, the command will fail unless the **-f** flag is specified, in which case an implicit unload will be performed first.

Flags

-disk <i>FileName</i>	Specifies the file name of the virtual media to load into the virtual optical device.
-f	Force the media to be loaded even if the virtual optical device already has media loaded.
-release	Forces the virtual optical device to be unlocked even if the client has a reserve on the device.
-vtd <i>VirtualTargetDevice</i>	The name of the virtual target device.

Examples

To load the virtual optical disk *clientData* into the virtual optical device *vopt1*, type the following command:

```
loadopt -disk clientData -vtd vopt1
```

loginmsg command

Purpose

Modifies the Virtual I/O Server partition's login herald.

Syntax

```
loginmsg { -reset | "Herald string" }
```

Description

The `loginmsg` command sets the Virtual I/O Server partition's login herald.

Flags

-reset Reset the login message back to the system default.

Examples

1. To set the login herald to `Welcome` followed by `login:` on a separate line, type:

```
loginmsg "Welcome\nlogin:"
```
2. To reset the login herald back to the system default, type:

```
loginmsg -reset
```

IVM `lpar_netboot` command

Purpose

Retrieves the media access control (MAC) address and physical location code from network adapters for a logical partition, or instructs a logical partition to do a network boot.

Syntax

To retrieve a MAC address:

```
lpar_netboot -M -n [-v] [-x] [-f] [-i] [-A] -t ent [-D -s Speed-d Duplex -S Server -G Gateway -C Client -K subnetmask] partition name partition profile manage system
```

To perform a network boot:

```
lpar_netboot [-v[-x] [-f] [-i] [-g args] [-A -D | [-D] -l phys_loc | [-D] -m maddress] -t ent [-D -s Speed-d Duplex -S Server -G Gateway -C Client -K subnetmask] partition name partition profile manage system
```

Description

The `lpar_netboot` command instructs a logical partition to do a network boot by having the logical partition send out a bootp request to a server specified with the **-S** flag. The server can be a network installation management (NIM) server serving SPOT resources or any server serving network boot images.

If the **-M** and **-n** flags are specified, the `lpar_netboot` command returns the MAC address and the physical location code for a particular type of network adapter specified with the **-t** flag. When the **-m** flag is specified, `lpar_netboot` boots a partition using a specific network adapter that matches the specified MAC address. When the **-l** flag is specified, `lpar_netboot` boots a logical partition using a specific physical location code for the network adapter that matches the specified physical location code. The matching MAC address or physical location code is dependent on the hardware resource allocation in the profile in which the logical partition was booted. The `lpar_netboot` command also requires arguments for the partition name and the partition profile (which contains the allocated hardware resources), and the name of the managed system in which the logical partition was defined.

Flags

-A	Returns all adapters of the given type.
-C Client	Specifies the IP address of the machine to do a network boot.
-D	Performs a ping test so the adapter can successfully ping the server that is specified with the -S flag.
-d Duplex	Specifies the duplex setting of the machine that is specified with the -C flag.
-f	Forces a close virtual terminal session for the logical partition.
-G Gateway	Specifies the gateway IP address of the machine that is specified with the -C flag.
-g args	Specifies generic arguments for booting.
-i	Forces an immediate shutdown of the partition.
-K subnetmask	Specifies the mask that the gateway should use in determining the appropriate subnetwork for routing. The subnet mask is a set of 4 bytes, as in the Internet address. The subnet mask consists of high bits (1s) corresponding to the bit positions of the network and subnetwork address, and low bits (0s) corresponding to the bit positions of the host address.
-l phys_loc	Specifies the physical location code of the network adapter to do a network boot.
-M	Displays the network adapter MAC address and physical location code.
-m maddress	Specifies the MAC address of the network adapter to do a network boot.
-n	Instructs the logical partition to not do a network boot.
-S Server	Specifies the IP address of the machine to retrieve the network boot image during the network boot.
-s Speed	Specifies the speed setting of the machine that is specified with the -C flag.
-t ent	Specifies the type of adapter for MAC address or physical location code discovery, or for a network boot.
-v	Displays additional information while the command is running.
-x	Displays debug output while the command is running.

Parameters

Parameter	Description
<i>partition_name</i>	Specifies the name of the partition.
<i>partition_profile</i>	Specifies the name of the partition profile to use.
<i>managed_system</i>	Specifies the name of the managed system on which the partition is defined.

Exit Status

The following exit values are returned:

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

Security

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

Examples

- To retrieve MAC address and physical location code for partition machA with logical partition profile machA_prof on managed system test_sys, type:

```
lpar_netboot -M -n -t ent "machA" "machA_prof" "test_sys"
```
- To do a network boot logical partition machA with a partition profile machA_prof on managed system test_sys, type:

```
lpar_netboot -t ent -s auto -d auto -S 9.3.6.49 -G 9.3.6.1 -C 9.3.6.234  
"machA" "machA_prof" "test_sys"
```

3. To do a network boot logical partition machA with the specific MAC address 00:09:6b:dd:02:e8 and logical partition profile machA_prof on managed system test_sys, type:

```
OS_install -o allocate -a os_resource=my53resource myclient01
```

4. To do a network boot logical partition machA with the specific physical location code U1234.121.A123456-P1-T6 and logical partition profile machA_prof on managed system test_sys, type:

```
lpar_netboot -t ent -l U1234.121.A123456-P1-T6 -s auto -d auto -S 9.3.6.49  
-G 9.3.6.1 -C 9.3.6.234 "machA" "machA_prof" "test_sys"
```

5. To perform a ping test and do a network boot of logical partition machA with logical partition profile machA_prof on a managed system test_sys, type:

```
lpar_netboot -t ent -D -s auto -d auto -S 9.3.6.49 -G 9.3.6.1 -C 9.3.6.234  
"machA" "machA_prof" "test_sys"
```

Related Information

The installios command.

IVM lpcfgop command

Purpose

Perform a partition configuration image operation. This command is valid only in an Integrated Virtualization Manager environment.

Syntax

To clear logical partition configuration data at next system restart:

```
lpcfgop -o clear [--force ] [ -m ManagedSystem ]
```

To disable logical partition configuration data at next system restart:

```
lpcfgop -o disable [--force ] [ -m ManagedSystem ]
```

To dump logical partition configuration data to a file:

```
lpcfgop -o dump [-f DumpFile] [ -m ManagedSystem ]
```

To enable logical partition configuration data at next system restart:

```
lpcfgop -o enable [ -m ManagedSystem ]
```

Description

The **lpcfgop** command performs a partition configuration image operation such as clearing, disabling, enabling, or dumping the logical partition configuration data.

Flags

-o <i>Operation</i>	The type of operations: <ul style="list-style-type: none">• clear: Marks the logical partition configuration data to be cleared when the managed system is restarted.• disable: Marks the logical partition configuration data to be disabled when the managed system is restarted. This will temporarily place the system back into the factory configuration partition mode.• dump: Dumps all logical partition configuration data from managed system firmware to a file. This data may be used by service. Use the bkprofddata command to create backups that can be restored.• enable: Enables partition configuration data so that it will not be disabled or cleared when the managed system is restarted.
-f <i>DumpFile</i>	The name of the file to write the dump to in the current working directory. If not specified, the default file will be /var/adm/lpm/lparConfig.dump
--force	Skip confirmation prompts for clear and disable operations.
-m <i>ManagedSystem</i>	The name of the managed system. This attribute is optional because there is only one system to manage. The name may either be the user-defined name for the managed system, or be in the form ttt-mmm*sssssss , where ttt is the machine type, mmm is the model, and sssssss is the serial number of the managed system.

Exit Status

This command has a return code of 0 on success.

Security

This command is not accessible by padmin.

Examples

1. Clear the partition configuration data (confirmation will be required) during the next reboot:

```
lpcfgop -o clear
```
2. Dump the logical partition configuration data to lparData.dump:

```
lpcfgop -o dump -f lparData.dump
```

Related Information

The **bkprofddata** command and the **rstprofddata** command.

ls command

Purpose

Displays the contents of a directory.

Syntax

To Display Contents of Directory or Name of File

```
ls [-1] [-A] [-C] [-F] [-H | -L] [-N] [-R] [-X] [-a] [-b] [-c] [-d] [-e] [-f] [-g] [-i] [-l] [-m] [-n] [-o] [-p] [-q] [-r] [-s] [-t] [-u] [-U] [-x] [File ...]
```

To Display Contents of Directory

```
ls -f [-C] [-d] [-i] [-m] [-s] [-X] [-x] [-1] [-U] [Directory ...]
```

Description

The **ls** command writes to standard output the contents of each specified *Directory* parameter or the name of each specified *File* parameter, along with any other information you ask for with the flags. If you do not specify a *File* or *Directory* parameter, the **ls** command displays the contents of the current directory.

Specifying more than one of the options in the mutually exclusive pairs is not considered an error. The last option specified in each pair determines the output format.

By default, the **ls** command displays all information in alphabetic order by file name. The collating sequence is determined by the **LANG** or **LC_COLLATE** environment variable.

When the **ls** command displays the contents of a directory, it does not show entries for files whose names begin with a . (dot) unless you use the **-a** or **-A** flag. If the command is executed by root, it uses the **-A** flag by default.

There are three main ways to format the output:

- List one entry per line.
- List entries in multiple columns by specifying either the **-C** or **-x** flag. The **-C** flag is the default format when output is to a tty. The **ls** command displays single column output if file or directory names are too long.
- List entries in a comma-separated series by specifying the **-m** flag.

To determine the number of character positions in the output line, the **ls** command uses the **COLUMNS** environment variable. If this variable is not set, the command gets the current column value of the display. If the **ls** command cannot determine the number of character positions by either of these methods, it uses a default value of 80.

The mode displayed with the **-U** flag is the same as with the **-l** flag, except for the addition of an 11th character interpreted as follows:

- E** Indicates a file has extended attributes (EA) information. The EA of a file is displayed by using the **getea** command.
- Indicates a file does not have extended attributes information.

The mode displayed with the **-e** and **-l** flags is interpreted as follows:

If the first character is:

- d** The entry is a directory.
- b** The entry is a block special file.
- c** The entry is a character special file.
- l** The entry is a symbolic link, and either the **-N** flag was specified or the symbolic link did not point to an existing file.
- p** The entry is a first-in,first-out (FIFO) special file.
- s** The entry is a local socket.
- The entry is an ordinary file.

The next nine characters are divided into three sets of three characters each. The first set of three characters show the owner's permission. The next set of three characters show the permission of the other users in the group. The last set of three characters shows the permission of anyone else with access to the file. The three characters in each set indicate, respectively, read, write, and execute permission of the file. Execute permission of a directory lets you search a directory for a specified file.

Permissions are indicated as follows:

r	Read
w	Write (edit)
x	Execute (search)
-	Corresponding permission not granted

The group-execute permission character is **s** if the file has set-group-ID mode. The user-execute permission character is **S** if the file has set-user-ID mode. The last character of the mode (usually **x** or **-**) is **T** if the 01000 (octal) bit of the mode is set (see the **chmod** command for the meaning of this mode). The indications of set-ID and 01000 bit of the mode are capitalized (**S** and **T**, respectively) if the corresponding execute permission is not set. The mode **t** indicates that the sticky bit is on for the file or the directory.

The mode displayed with the **-e** flag is the same as with the **-l** flag, except for the addition of an 11th character interpreted as follows:

+	Indicates a file has extended security information. For example, the file may have extended ACL , TCB , or TP attributes in the mode.
-	Indicates a file does not have extended security information.

When the size of the files in a directory are listed, the **ls** command displays a total count of blocks, including indirect blocks.

Flags

-A	Lists all entries except . (dot) and .. (dot-dot).
-a	Lists all entries in the directory, including the entries that begin with a . (dot).
-b	Displays nonprintable characters in an octal (\nnm) notation.
-c	Uses the time of last modification of the i -node for either sorting (when used with the -t flag) or for displaying (when used with the -l flag). This flag must be used with either the -t or -l flag, or both.
-C	Sorts output vertically in a multicolumn format. This is the default method when output is to a terminal.
-d	Displays only the information for the directory named. Directories are treated like files, which is helpful when using the -l flag to get the status of a directory.
-e	Displays the mode (including security information), number of links, owner, group, size (in bytes), time of last modification, and name of each file. If the file is a special file, the size field contains the major and minor device numbers. If the file is a symbolic link, the path name of the linked-to file is printed preceded by a -> (minus, greater than) sign. The attributes of the symbolic link are displayed.
-E	Lists space reservation, fixed extent size, and extent allocation flag information for a file. -l must be specified with this flag.
-f	Lists the name in each slot for each directory specified in the <i>Directory</i> parameter. This flag turns off the -l , -t , -s , and -r flags, and turns on the -a flag. The order of the listing is the order in which entries appear in the directory.
-F	Puts a / (slash) after each file name if the file is a directory, an * (asterisk) if the file can be executed, an = (equal sign) if the file is a socket, a (pipe) sign if the file is a FIFO, and an @ for a symbolic link. Note: Symbolic links are displayed with the trailing -> only if the -N flag is used or if the link points to a nonexistent file. Otherwise, information about the target file is displayed. You can also invoke this option by entering the ls -f command.
-g	Displays the same information as the -l flag, except the -g flag suppresses display of the owner and symbolic link information.
-H	If a symbolic link referencing a file of type directory is specified on the command line, the ls command shall evaluate the file information and file type to be those of the file referenced by the link, and not the link itself; however, the ls command shall write the name of the link itself and not the file referenced by the link.
-i	Displays the i -node number in the first column of the report for each file.

- L** Lists the file or directory contents that the link references. This is the default action. Symbolic links are followed. If the **-l** option is used, the **-N** option becomes the default, and no symbolic links are followed. When the **-l** option is used, only the **-L** option can override the **-N** default.
- l** (Lower case L) Displays the mode, number of links, owner, group, size (in bytes), and time of last modification for each file. If the file is a special file, the size field contains the major and minor device numbers. If the time of last modification is greater than six months ago, the time field is shown in the format **month date year** where as files modified within six months the time field is shown as **month date time** format.

If the file is a symbolic link, the path name of the linked-to file is printed preceded by a **->**. The attributes of the symbolic link are displayed. The **-n**, **-g**, and **-o** flag overrides the **-l** flag.
Notes:
 1. A symbolically linked file is followed by an arrow and the contents of the symbolic link.
 2. The performance of the **ls** command when used with the **-l** option can be improved by executing the **mkpasswd** command. This is helpful when a directory contains files owned by different users, such as the **/tmp** directory.
- m** Uses stream output format (a comma-separated series).
- n** Displays the same information as the **-l** flag, except that the **-n** flag displays the user and the group IDs instead of the user and group names.
- N** Does not follow symbolic links when determining the status of a file.
Note: If both the **-L** and **-N** options are used, the last one will dominate. Also, any time a symbolic link is given that includes a **/** (slash) as the final character, the link will automatically be followed regardless of any options used.
- o** Displays the same information as the **-l** flag, except the **-o** flag suppresses display of the group and symbolic link information.
- p** Puts a slash after each file name if that file is a directory. This is useful when you pipe the output of the **ls** command to the **pr** command, as follows:

```
ls -p | pr -5 -t -w80
```
- q** Displays nonprintable characters in file names as a **?** (question mark).
- r** Reverses the order of the sort, giving reverse alphabetic or the oldest first, as appropriate.
- R** Lists all subdirectories recursively.
- s** Gives size in kilobytes (including indirect blocks) for each entry.
- t** Sorts by time of last modification (latest first) instead of by name.
- U** Displays similar information as the **-l** flag. Displays the mode (including security information and named extended attribute information), number of links, owner, group, size (in bytes), time of last modification, and name of each file. If the file is a special file, the size field contains the major and minor device numbers. If the file is a symbolic link, the path name of the linked-to file is printed preceded by a **->** (minus, greater than) sign. The attributes of the symbolic link are displayed.
- u** Uses the time of the last access, instead of the time of the last modification, for either sorting (when used with the **-t** flag) or for displaying (when used with the **-l** flag). This flag has no effect if it is not used with either the **-t** or **-l** flag, or both.
- x** Sorts output horizontally in a multi-column format.
- X** Prints long user names when used with other flags that display user names. The upper limit is determined by the **max_logname** ODM attribute in the PdAt and CuAt object classes. If a user name is greater than the **max_logname** attribute, it will be truncated to the number of characters as specified by the **max_logname** attribute, less one character.
- 1** Forces output into one-entry-per-line format. This is the default when the output is not directed to a terminal.

Exit Status

This command returns the following exit values:

- 0 All files were written successfully.
- >0 An error occurred.

Examples

1. To list all files in the current directory, type:

```
ls -a
```

This lists all files, including . (dot), .. (dot-dot), and other files with names beginning with a dot.

2. To display detailed information, type:

```
ls -l chap1 .profile
```

This displays a long listing with detailed information about chap1 and .profile.

3. To display detailed information about a directory, type:

```
ls -d -l .manual manual/chap1
```

This displays a long listing for the directories . and manual, and for the file manual/chap1. Without the **-d** flag, this would list the files in the . and manual directories instead of the detailed information about the directories themselves.

4. To list the files in order of modification time, type:

```
ls -l -t
```

This displays a long listing of the files that were modified most recently, followed by the older files.

5. To display detailed information with expanded user and group name, type:

```
ls -lX .profile
```

This displays a long listing with detailed information about **.profile**.

6. To display about whether extended attributes are set on the files in current directory, type:

```
ls -U
```

Example output:

```
-rwSr-x---+ 1 root    system      28 Apr 29 03:23 only_aixc
-rwSr-x---E 1 root    system       4 Apr 29 03:23 only_aixc_ea
-rw-r--r--E 1 root    system       4 Apr 29 03:23 only_ea
-----+    1 root    system     265 Apr 29 03:23 only_nfs4
-----E    1 root    system     64 Apr 29 03:23 only_nfs4_ea
-rw-r--r--- 1 root    system       4 Apr 29 03:23 only_regular
```

Files

<code>/usr/bin/ls</code>	Contains the <code>ls</code> command.
<code>/etc/passwd</code>	Contains user IDs.
<code>/etc/group</code>	Contains group IDs.
<code>/usr/share/lib/terminfo/*</code>	Contains terminal information.

Related Information

The `chmod` command.

lsdev command

Purpose

Displays Virtual I/O Server devices and their characteristics.

Syntax

To list devices:

```
lsdev [ -type DeviceType... ] [ -virtual ] [ -field FieldName... ] [ -fmt Delimiter ] [ -state State ]
```

To display information about a specific device:

```
lsdev { -dev DeviceName | -plc PhysicalLocationCode } [ -child ] [ -field FieldName... ] [ -fmt Delimiter ]
```

```
lsdev { -dev DeviceName | -plc PhysicalLocationCode } [ -attr [ Attribute ] | -range Attribute | -slot | -vpd | -parent ]
```

```
lsdev -vpd
```

```
lsdev -slots
```

Description

The **lsdev** command displays information about devices in the Virtual I/O Server. If no flags are specified, a list of all devices, both physical and virtual, in the Virtual I/O Server is displayed. To list devices, both physical and virtual, of a specific type use the **-type** *DeviceType* flag. Use the **-virtual** flag to list only virtual devices. Combining both the **-type** and **-virtual** flags will list the virtual devices of the specified type.

To display information about a specific device, use the **-dev** *DeviceName* or **-plc** *PhysicalLocationCode*. Use either the **-child**, **-parent**, **-attr**, **-range**, **-slot**, or **-vpd** flag to specify what type of information is displayed. If none of these flags are used, the name, status, and description of the device will be displayed.

Using the **-vpd** flag, without specifying a device, displays platform-specific information for all devices.

If the **-fmt** *Delimiter* flag is specified the **lsdev** command returns all output in a delimiter separated format. The **-state** *State* flag limits the output to devices in the indicated state. The **-slots** flag produces a list of I/O slot information for built-in adapters that are not hot-pluggable but can have dynamic LPAR operations performed on them.

The **lsdev** output will be truncated to 80 characters per line, unless the **-fmt** flag is used. If the **-fmt** flag is used, the output lines will not be truncated and line wrap may occur, based on what the terminal window line length is set to.

Flags

- | | |
|-----------------------------------|---|
| -attr [<i>Attribute</i>] | Displays information about attributes of a given device. If an attribute is specified, its current value is displayed. If no attribute is specified, the following information is displayed about all attributes for the give device:

attribute
Attribute name

value Current value of the attribute

description
Description of the attribute

user Whether attribute can be set by the user (TRUE/FALSE) |
| -child | Displays the name, status, physical location code, and description for each child of the specified device (-dev <i>DeviceName</i> or -plc <i>PhysicalLocationCode</i>). |
| -dev <i>DeviceName</i> | Specifies the device logical name for the device for which information is listed. This flag cannot be used with the -plc flag. |

-field <i>FieldName</i>	<p>Specifies the list of fields to be displayed. The following fields are supported:</p> <p>name Device name</p> <p>status Device status</p> <p>physloc Physical location code</p> <p>description Description of the device</p> <p>parent Note: The -field flag cannot be combined with the -parent, -attr, -range, -slot, or -vpd flags.</p>
-fmt <i>Delimiter</i>	Specifies a delimiter character to separate output fields.
-parent	Displays the name, status, physical location code, and description of the parent device of the given device (-dev <i>DeviceName</i> or -plc <i>PhysicalLocationCode</i>).
-plc <i>PhysicalLocationCode</i>	Specifies the device physical location code for the device for which information is listed. This flag cannot be used with the -dev flag.
-range <i>Attribute</i>	Displays the allowed values for the specified attribute.
-slot	Displays the slot, description, and device name of the specified device (-dev <i>DeviceName</i> or -plc <i>PhysicalLocationCode</i>). The device must be in a PCI hot swappable slot.
-slots	Displays a list of I/O slot information for built-in adapters that are not hot-pluggable but can have dynamic LPAR operations performed on them.
-State <i>State</i>	<p>Limits the output to devices in the specified state. The following states are supported:</p> <p>0, defined Server Virtual Adapter.</p> <p>1, available Server Virtual Adapter Physical Location Code</p> <p>2, stopped Client Partition ID</p>
-type <i>DeviceType</i>	<p>Specifies the device type. This flag can be used to restrict output to devices in the specified types. Both physical and virtual devices are listed.</p> <p>Supported types are as follows:</p> <p>adapter Lists adapters</p> <p>disk Lists disks</p> <p>lv Lists logical volumes and volume groups</p> <p>optical Lists optical devices (cdrom/dvdrom)</p> <p>tape Lists tape devices</p> <p>tape4vtd Lists tape devices available for creating virtual target devices</p> <p>tty Lists tty devices</p> <p>ent4sea Lists all physical Ethernet adapters and Etherchannel adapters available for creating a shared Ethernet adapter</p> <p>ven4sea Lists all virtual Ethernet adapters available for creating shared Ethernet adapter</p> <p>ent4ip Lists all adapters over which interface can be configured.</p>
-virtual	Restricts output to virtual devices only.
-vpd	Displays platform-specific information for all devices or for a single device when -dev <i>DeviceName</i> or -plc <i>PhysicalLocationCode</i> are specified.

Exit Status

Table 1. Command specific return codes

Return code	Description
12	Specified logical volume belongs to the operating system.
13	Specified physical or logical volume is not valid physical or logical volume

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

1. To list all virtual adapters and display the **name** and **status** fields, type:

```
lsdev -type adapter -virtual -field name status
```

The system displays a message similar to the following:

```
name status
```

```
vhost0 Available
vhost1 Available
vhost2 Available
ent6 Available
ent7 Available
ent8 Available
ent9 Available
```

2. To list all devices of type **disk** and display the name and physical location fields, type:

```
lsdev -type disk -field name physloc
```

The system displays a message similar to the following:

```
name physloc
```

```
hdisk0 U9111.520.10004BA-T15-L5-L0
hdisk1 U9111.520.10004BA-T15-L8-L0
hdisk2 U9111.520.10004BA-T16-L5-L0
hdisk3 U9111.520.10004BA-T16-L8-L0
hdisk4 UTMP0.02E.00004BA-P1-C4-T1-L8-L0
hdisk5 UTMP0.02E.00004BA-P1-C4-T2-L8-L0
hdisk6 UTMP0.02F.00004BA-P1-C8-T2-L8-L0
hdisk7 UTMP0.02F.00004BA-P1-C4-T2-L8-L0
hdisk8 UTMP0.02F.00004BA-P1-C4-T2-L11-L0
vtscsi0 U9111.520.10004BA-V1-C2-L1
vtscsi1 U9111.520.10004BA-V1-C3-L1
vtscsi2 U9111.520.10004BA-V1-C3-L2
vtscsi3 U9111.520.10004BA-V1-C4-L1
vtscsi4 U9111.520.10004BA-V1-C4-L2
vtscsi5 U9111.520.10004BA-V1-C5-L1
```

3. To display the parent of a devices, type:

```
lsdev -dev hdisk0 -parent
```

The system displays a message similar to the following:

```
parent
```

```
scsi0
```

4. To display all I/O slots that are not hot-pluggable but can have DLPAR operations performed on them, type:

```
lsdev -slots
```

The system displays a message similar to the following:

```
U787A.001.DNZ00Y1-P1-C1 Logical I/O Slot pci4 sisscsia0
U787A.001.DNZ00Y1-P1-T5 Logical I/O Slot pci3 ent0 ent1
U787A.001.DNZ00Y1-P1-T7 Logical I/O Slot pci2 usbhc0 usbhc1
U9111.520.10DFD8C-V2-C0 Virtual I/O Slot vsa0
U9111.520.10DFD8C-V2-C2 Virtual I/O Slot vhost0
U9111.520.10DFD8C-V2-C4 Virtual I/O Slot Unknown
```

Related Information

The `cfgdev` command, the `chdev` command, the `chpath` command, the `lsmmap` command, the `lspath` command, the `mkpath` command, the `mkvdev` command, the `rmdev` command, and the `rmpath` command.

lsfailedlogin command

Purpose

Lists the contents of the failed login log to the screen.

Syntax

```
lsfailedlogin
```

Description

The `lsfailedlogin` command dumps the contents of the failed login log. The failed login file records unsuccessful login attempts by any user on the Virtual I/O Server.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Security

This command can only be executed by the prime administrator (`padmin`) user.

Examples

1. To list all failed logins, type:

```
lsfailedlogin
```

Related Information

The `lsgcl` command.

lsfware command

Purpose

Displays microcode and firmware levels of the system, adapters and devices.

Syntax

```
lsfware [ -all | -dev Name ]
```

Description

The **lsfware** command displays the platform system firmware microcode level and the service processor microcode levels, if supported. Not all systems contain a service processor, nor do all systems support displaying the system processor level. Information on a specific device is displayed with the **-dev** flag.

The **-all** flag displays system firmware/microcode for all devices. The output from the **-all** flag is always delimiter separated.

Flags

-all Displays microcode level information for all supported devices.
-dev Name Displays microcode level information for the named device.

Examples

1. To display the system firmware level and service processor (if present), type:

```
lsfware
```

The system displays a message similar to the following:

```
System Firmware level is TCP99256
```

2. To display the microcode level for all supported devices, type:

```
lsfware -all
```

The system displays a message similar to the following:

```
sys0|system:TCP99256
rmt0|C009
scraid0|adapter:4.20.18|adapter-boot:4.00.26
raid-dasd|22:FFC #:DDYS-T0.524D3031.53393446
raid-dasd|26:FFC #:DDYS-T0.524D3031.53393446
raid-dasd|2e:FFC #:DDYS-T0.525A3034.53393243
....
```

Isgcl command

Purpose

Lists the contents of the global command log to the screen.

Syntax

lsycl

Description

The **Isgcl** command lists the contents of the global command log (gcl). This log contains a listing of all commands that have been executed by all Virtual I/O Server users. Each listing contains the date and time of execution as well as the userid the command was executed from.

Global command log file format is as follows:

Date	Time	userid	Command	Command options
mmm dd yyyy	hh:mm:ss	ccccccc	Command	Command options span to 80 characters then wraps to the next row

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Security

This command can only be executed by the prime administrator (padmin) user.

Examples

1. To list the contents of the global command log, type:

```
lsgcl
```

Related Information

The lsfailedlogin command.

IVM lshwres command

Purpose

Lists the hardware resources of a managed system. This command is valid only in an Integrated Virtualization Manager environment.

Syntax

To list attributes for physical I/O buses

```
lshwres -r io --rsubtype bus [ --filter "FilterData" ] [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

To list attributes for I/O pools

```
lshwres -r io --rsubtype iopool --level pool [ --filter "FilterData" ] [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

To list system attributes for I/O pools

```
lshwres -r io --rsubtype iopool --level sys [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

To list attributes for physical I/O buses

```
lshwres -r io --rsubtype bus [ --filter "FilterData" ] [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

To list attributes for physical I/O slots

```
lshwres -r io --rsubtype slot [ --filter "FilterData" ] [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

To list attributes for physical I/O units

```
lshwres -r io --rsubtype unit [ --filter "FilterData" ] [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```


To list recoverable I/O resources

```
lshwres -r io --rsubtype slot -R [ --filter "FilterData" ] [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

To list the currently tagged I/O for all IBM i logical partitions

```
lshwres -r io --rsubtype taggedio [ --filter "FilterData" ] [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

To list partition attributes for memory

```
lshwres -r mem --level lpar [ --filter "FilterData" ] [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

To display memory statistics from a partition

```
lshwres -r mem --level lpar --stat --filter "FilterData" [-F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

To list system attributes for memory

```
lshwres -r mem --level sys [ --maxmem ] [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

To list recoverable memory resources

```
lshwres -r mem --level lpar -R [ --filter "FilterData" ] [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

To list information about the memory pool

```
lshwres -r mempool [ --filter "FilterData" ] [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

To list paging devices associated with the memory pool

```
lshwres -r mempool --rsubtype pgdev [ --filter "FilterData" ] [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

To list recoverable memory pools

```
lshwres -r mempool -R [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

To list partition attributes for processors

```
lshwres -r proc --level lpar [ --filter "FilterData" ] [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

To list shared processor pool attributes

```
lshwres -r proc --level pool [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

To list system attributes for processors

```
lshwres -r proc --level sys [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

To list recoverable processors resources

lshwres -r proc --level lpar -R [--filter "FilterData"] [-F "AttributeNames"] [--header] [-m ManagedSystem]

To list virtual Ethernet adapter attributes

lshwres -r virtualio --rsubtype eth --level lpar [--filter "FilterData"] [-F "AttributeNames"] [--header] [-m ManagedSystem]

To list system attributes for virtual Ethernet adapters

lshwres -r virtualio --rsubtype eth --level sys [-F "AttributeNames"] [--header] [-m ManagedSystem]

To list virtual fibre channel adapter attributes

lshwres -r virtualio --rsubtype fc --level lpar [--filter "FilterData"] [-F "AttributeNames"] [--header] [-m ManagedSystem]

To list system attributes for virtual fibre channel adapters

lshwres -r virtualio --rsubtype fc --level sys [-F "AttributeNames"] [--header] [-m ManagedSystem]

To list virtual SCSI adapter attributes

lshwres -r virtualio --rsubtype scsi [--level lpar] [--filter "FilterData"] [-F "AttributeNames"] [--header] [-m ManagedSystem]

To list virtual serial adapter attributes

lshwres -r virtualio --rsubtype serial --level lpar [--filter "FilterData"] [-F "AttributeNames"] [--header] [-m ManagedSystem]

To list partition attributes for virtual I/O slots

lshwres -r virtualio --rsubtype slot --level lpar [--filter "FilterData"] [-F "AttributeNames"] [--header] [-m ManagedSystem]

To list virtual I/O slot attributes

lshwres -r virtualio --rsubtype slot --level slot [--filter "FilterData"] [-F "AttributeNames"] [--header] [-m ManagedSystem]

To list recoverable virtual I/O resources

lshwres -r virtualio --rsubtype slot -- level slot -R [--filter "FilterData"] [-F "AttributeNames"] [--header] [-m ManagedSystem]

To list high-speed link (HSL) pools for all IBM i logical partitions

lshwres -r virtualio --rsubtype hsl [--filter "FilterData"] [-F "AttributeNames"] [--header] [-m ManagedSystem]

To list Virtual OptiConnect pools for all IBM i logical partitions

lshwres -r virtualio --rsubtype virtualopti [--filter "FilterData"] [-F "AttributeNames"] [--header] [-m ManagedSystem]

To list the physical Host Ethernet Adapters

```
lshwres -r hea --rsubtype phys --level sys [ --filter "FilterData" ] [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

To list the physical Host Ethernet Adapters ports

```
lshwres -r hea --rsubtype phys --level port [ --filter "FilterData" ] [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

To list the physical Host Ethernet Adapters port performance metrics

```
lshwres -r hea --rsubtype phys --level port --stat [ --filter "FilterData" ] [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

To list the physical Host Ethernet Adapters port groups

```
lshwres -r hea --rsubtype phys --level port_group [ --filter "FilterData" ] [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

To list the logical Host Ethernet Adapters and resource settings

```
lshwres -r hea --rsubtype logical --level sys [ --filter "FilterData" ] [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

To list the logical ports

```
lshwres -r hea --rsubtype logical --level port [-R] [ --filter "FilterData" ] [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

Description

The **lshwres** command lists the hardware resources of the managed-system, including physical I/O, virtual I/O, memory, and processing.

Flags

-r <i>ResourceType</i>	The combination of -r , --rsubtype , and --level indicates which type of attributes and objects to list. The valid combinations are enumerated below.
--rsubtype	
<i>ResourceSubtype</i> --level	
<i>ResourceLevel</i>	

- **-r io --rsubtype bus**: List attributes for physical I/O buses
 - Attributes: unit_phys_loc,bus_id,backplane_phys_loc,bus_drc_index,bus_drc_name
 - Filters: units, buses

- **-r io --subtype iopool -level sys:** List attributes for physical I/O pools
 - Attributes: max_io_pools
 - Filters: none
- **-r io --subtype iopool -level pool:** List attributes for physical I/O pools
 - Attributes: io_pool_id, lpar_ids, slots
 - Filters: {lpar_ids | lpar_names}, pools
- **-r io --subtype slot:** List attributes for physical I/O slots
 - Attributes: unit_phys_loc, bus_id, phys_loc, drc_index, lpar_name, lpar_id, slot_io_pool_id, description, feature_codes, adapter_feature_codes, adapter_descriptions, pci_vendor_id, pci_device_id, pci_subs_vendor_id, pci_subs_device_id, pci_class, pci_revision_id, bus_grouping, iop, iop_info_stale, console_capable, load_source_capable, load_source_attached, alt_restart_device_capable, alt_restart_device_attached, op_console_capable, op_console_attached, twinax_capable, twinax_attached, direct_console_capable, lan_console_capable, vpd_stale, vpd_type, vpd_model, vpd_serial_num, parent_slot_drc_index, drc_name
 - Filters: {lpar_ids | lpar_names}, units, buses, slots, pools
- **-r io --subtype slot -R:** List attributes for physical I/O slots
 - Attributes: drc_index, lpar_name, lpar_id, drc_name
 - Filters: {lpar_ids | lpar_names}, units, buses, slots, pools
- **-r io --subtype taggedio:** List attributes for tagged I/O resources
 - Attributes: lpar_name, lpar_id, load_source_slot, alt_restart_device_slot, recent_alt_restart_device_slot, console_slot, alt_console_slot, op_console_slot
 - Filters: {lpar_ids | lpar_names}
- **-r io --subtype unit:** List attributes for physical I/O units
 - Attributes: unit_phys_loc
 - Filters: units
- **-r hea --subtype phys --level sys**
 - Attributes: adapter_id, state, phys_loc
 - Filters: adapter_ids
- **-r hea --subtype phys --level port_group**
 - Attributes: adapter_id, port_group, phys_port_ids, unassigned_logical_port_ids, curr_port_group_mcs_value, pend_port_group_mcs_value, valid_port_group_mcs_values
 - Filters: adapter_ids, port_groups
- **-r hea --subtype phys --level port**
 - Attributes: adapter_id, port_group, phys_port_id, phys_port_type, phys_port_state, conn_state, curr_conn_speed, config_conn_speed, curr_duplex, config_duplex, trans_flow_control, recv_flow_control, config_flow_control, max_recv_packet_size, promisc_lpar_id, promisc_lpar_name, logical_port_ids, phys_port_loc
 - Filters: adapter_ids, port_groups

- **-r hea --subtype phys --level port --stat**
 - Attributes: adapter_id, port_group, phys_port_id, recv_octets, recv_packets_0_64, recv_packets_65_127, recv_packets_128_255, recv_packets_256_511, recv_packets_512_1023, recv_packets_1024_max, recv_packets_dropped_bad_FCS, recv_packets_dropped_internal_mac_error, recv_packets_dropped_in_range_length_error, recv_packets_dropped_out_of_range_length_error, recv_packets_dropped_frame_too_long, recv_packets_dropped_jabber, recv_symbol_error, recv_code_error, recv_runt_frame, recv_fragments, recv_unsupported_opcode, recv_control_pause_frames, recv_search_busy, recv_packets_dropped_filter, recv_packets_dropped_other, recv_packets_dropped_alignment, recv_MC_packets, recv_BC_packets, trans_MC_packets, trans_BC_packets, trans_octets, trans_packets_length_0_64, trans_packets_length_65_127, trans_packets_length_128_255, trans_packets_length_255_511, trans_packets_length_512_1023, trans_packets_length_1024_max, trans_packets_dropped_bad_FCS, trans_control_pause_frames, trans_tx_local_fault_packets, trans_tx_remote_fault_packets, trans_tx_packets_dropped_int_MAC_error, trans_packets_retried_single_collision, trans_packets_retried_multiple_collision, trans_packets_signal_quality, trans_packets_deferred, trans_packets_late_collisions, trans_packets_excessive_collisions, trans_packets_no_carrier, recv_overrun_frames_dropped
 - Filters: adapter_ids, port_groups
- **-r hea --subtype logical --level sys**
 - Attributes: adapter_id, lpar_id, lpar_name, drc_index, drc_name, capabilities, ieq, nieq, qp, cq, mr
 - Filters: {lpar_ids | lpar_names}, adapter_ids
- **-r hea --subtype logical --level port**
 - Attributes: adapter_id, lpar_id, lpar_name, state, port_group, phys_port_id, logical_port_id, drc_index, drc_name, mac_addr, user_def_mac_addr, vlan_id_list
 - Filters: {lpar_ids | lpar_names}, adapter_ids, port_groups
- **-r mem --level lpar:** List partition attributes for memory
 - Attributes: lpar_name, lpar_id, mem_mode, auto_io_entitled_mem, curr_io_entitled_mem, curr_min_mem, curr_mem, curr_mem_weight, curr_max_mem, pend_io_entitled_mem, pend_min_mem, pend_mem, pend_mem_weight, pend_max_mem, run_io_entitled_mem, run_mem_weight, run_min_mem, run_mem
 - Filters: {lpar_ids | lpar_names}
- **-r mem --level lpar -R:** List partition attributes for memory
 - Attributes: lpar_name, lpar_id, mem_mode, pend_mem_weight, pend_io_entitled_mem, run_mem_weight, run_io_entitled_mempend_mem, run_mem
 - Filters: {lpar_ids | lpar_names}
- **-r mem --level lpar -stat:** List partition attributes for memory
 - Attributes: min_io_entitled_mem, optimial_io_entitled_mem, max_io_entitled_mem_usage
 - Filters: {lpar_ids | lpar_names}

- **-r mem --level sys [--maxmem]:** List system attributes for memory
 - Attributes: configurable_sys_mem, curr_avail_sys_mem, pend_avail_sys_mem, installed_sys_mem, deconfig_sys_mem, sys_firmware_mem, mem_region_size, pend_mem_region_size, possible_mem_region_size, max_mem_pools, max_paging_vios_per_mem_pool
 - Additional attributes with --maxmem: required_min_mem_aix_linux, required_min_mem_os400
 - Filters: None
- **-r mempool :** List system attributes for memory pools
 - Attributes: curr_pool_mem, pend_pool_mem, curr_max_pool_mem, pend_max_pool_mem, sys_firmware_pool_mem, paging_storage_pool, paging_vios_names, paging_vios_ids
 - Filters: None
- **-r mempool -R:** List system attributes for recoverable memory pool resources
 - Attributes: curr_pool_mem, pend_pool_mem, curr_max_pool_mem, pend_max_pool_mem, sys_firmware_pool_mem, paging_storage_pool, paging_vios_names, paging_vios_ids
 - Filters: None
- **-r mempool --subtype pgdev:** List paging devices associated with the memory pool
 - Attributes: device_name, size, state, type, phys_loc, storage_pool, lpar_name, lpar_id, paging_vios_name, paging_vios_id, redundant_state, is_redundant, redundant_device_name, redundant_paging_vios_id, redundant_paging_vios_name, redundant_phys_loc
 - Filters: {lpar_ids | lpar_names}
- **-r proc --level lpar:** List partition attributes for processors
 - Attributes: lpar_name, lpar_id, curr_shared_proc_pool_id, curr_proc_mode, curr_min_proc_units, curr_proc_units, curr_max_proc_units, curr_min_procs, curr_procs, curr_max_procs, curr_sharing_mode, curr_uncap_weight, pend_shared_proc_pool_id, pend_proc_mode, pend_min_proc_units, pend_proc_units, pend_max_proc_units, pend_min_procs, pend_procs, pend_max_procs, pend_sharing_mode, pend_uncap_weight, run_proc_units, run_procs, run_uncap_weight
 - Filters: {lpar_ids | lpar_names}
- **-r proc --level lpar -R:** List partition attributes for processors
 - Attributes: lpar_name, lpar_id, curr_proc_mode, curr_sharing_mode, pend_proc_mode, pend_proc_units, pend_procs, pend_sharing_mode, pend_uncap_weight, run_proc_units, run_procs, run_uncap_weight
 - Filters: {lpar_ids | lpar_names}
- **-r proc --level pool:** List shared processor pool attributes
 - Attributes: shared_proc_pool_id, configurable_pool_proc_units, curr_avail_pool_proc_units, pend_avail_pool_proc_units
 - Filters: None
- **-r proc --level sys:** List system attributes for processors
 - Attributes: configurable_sys_proc_units, curr_avail_sys_proc_units, pend_avail_sys_proc_units, installed_sys_proc_units, deconfig_sys_proc_units, min_proc_units_per_virtual_proc, max_shared_proc_pools, max_virtual_procs_per_lpar, max_procs_per_lpar
 - Filters: None
- **-r virtualio --subtype eth --level lpar:** List virtual Ethernet adapter attributes
 - Attributes: lpar_name, lpar_id, slot_num, state, ieee_virtual_eth, port_vlan_id, addl_vlan_ids, is_trunk, trunk_priority, is_required, mac_addr
 - Filters: {lpar_ids | lpar_names}, vlans, slots

- **-r virtualio --subtype eth --level sys:** List system attributes for virtual ethernet adapters
 - Attributes: max_vlans_per_port, mac_prefix
 - Filters: None
- **-r virtualio --subtype fc--level lpar:** List information about virtual fibre channel adapters for each logical partition
 - Attributes: lpar_name, lpar_id, slot_num, adapter_type, remote_lpar_id, remote_lpar_name, remote_slot_num, is_required, wwpns, state
 - Filters: {lpar_ids | lpar_names}, slots
- **-r virtualio --subtype fc--level sys:** List system attributes for virtual fibre channel adapters
 - Attributes: num_wwpns_remaining, wwpn_prefix
 - Filters: None
- **-r virtualio --subtype hsl:** List system attributes for high-speed link
 - Attributes: hsl_pool_id, lpar_names, lpar_ids
 - Filters: {lpar_ids | lpar_names}, pools
- **-r virtualio --subtype scsi --level lpar:** List virtual SCSI adapter attributes
 - Attributes: lpar_name, lpar_id, slot_num, state, adapter_type, remote_lpar_id, remote_lpar_name, remote_slot_num, is_required
 - Filters: {lpar_ids | lpar_names}, slots
- **-r virtualio --subtype scsi --level lpar:** List virtual SCSI adapter attributes
 - Attributes: vd_name, vscsi_client_name, vscsi_server_name, bd_sp_name, bd_name
 - Filters: {lpar_ids | lpar_names}, slots
- **-r virtualio --subtype slot --level lpar:** List partition attributes for virtual I/O slots
 - Attributes: lpar_name, lpar_id, curr_max_virtual_slots, pend_max_virtual_slots
 - Filters: {lpar_ids | lpar_names}
- **-r virtualio --subtype slot --level slot:** List virtual I/O slot attributes
 - Attributes: slot_num, lpar_name, lpar_id, config, state, drc_name
 - Filters: {lpar_ids | lpar_names}, slots
- - **-r virtualio --subtype slot --level slot -R:** List virtual I/O slot attributes
 - Attributes: slot_num, lpar_name, lpar_id, drc_name
 - Filters: {lpar_ids | lpar_names}, slots
- **-r virtualio --subtype virtualopti:** List virtual OptiConnect resource attributes
 - Attributes: virtual_opti_pool_id, lpar_names, lpar_ids
 - Filters: {lpar_ids | lpar_names}, pools

--maxmem
MaximumMemory

When this option is specified, the required minimum memory amount needed for partitions to support the maximum memory quantity specified is listed. All memory quantities are in megabytes, and are a multiple of the memory region size for the managed- system.

This information is useful for specifying memory amounts in partition profiles.

-m *ManagedSystem*

This option is only valid when listing system level memory resources.

The name of the managed system. This attribute is optional because there is only one system to manage. The name may either be the user-defined name for the managed system, or be in the form tttt-mmm*sssssss, where tttt is the machine type, mmm is the model, and sssssss is the serial number of the managed system.

--filter *FilterData*

The filters to apply to the resources to be listed. Filters are used to select which resources of the specified resource type are to be listed. If no filters are used, then all of the resources of the specified resource type will be listed. For example, specific partitions can be listed by using a filter to specify the names or IDs of the partitions to list. Otherwise, if no filter is used, then all the partitions in the managed system will be listed.

The filter data consists of filter name/value pairs, which are in comma separated value (CSV) format. The filter data must be enclosed in double quotation marks.

The format of the filter data is as follows:

```
"filter-name=value,filter-name=value,..."
```

Note that certain filters accept a comma separated list of values, as follows:

```
"filter-name=value,value,...",..."
```

When a list of values is specified, the filter name/value pair must be enclosed in double quotation marks. Depending on the shell being used, nested double quotation marks characters might need to be preceded by an escape character, which is usually a `'\'` character.

Unless otherwise indicated, multiple values can be specified for each filter.

Valid filter names:

adapter_ids

The DRC index of the Host Ethernet Adapter in hexadecimal

buses The bus ID of the I/O bus to view

lpar_ids | lpar_names

Name or ID of the logical partition to view

Note: The filter value of none for lpar_ids is supported and displays only paging devices that are not currently assigned to any logical partition. If the memory pool does not exist, and you use the `-r mempool --rsubtype pgdev` attribute, an error is displayed.

pools The pool ID of the I/O pool to view

port_groups

The Host Ethernet Adapter port group or groups

slots For physical I/O slots, the DRC index of the slot to view. For virtual I/O slots, the virtual slot number of the slot to view

units The physical location code of the unit to view

v lans The virtual LAN of the virtual Ethernet adapters to view

-F *AttributeNames*

A delimiter separated list of attribute names for the desired attribute values to be displayed for each resource. If no attribute names are specified, then values for all of the attributes for the resource will be displayed.

When this option is specified, only attribute values will be displayed. No attribute names will be displayed. The attribute values displayed will be separated by the delimiter which was specified with this option.

This option is useful when only attribute values are desired to be displayed, or when the values of only selected attributes are desired to be displayed.

Attribute names:

adapter_descriptions

Indicates a description of the adapter.

adapter_feature_codes

List of possible feature codes for the I/O adapter sorted by likely match. Each element in the list consists of two feature codes - one for each operating system. This list follows the following format:

```
aix_feature_code1/linux_feature_code1,  
/i5_feature_code1,aix_feature_code2/  
linux_feature_code2/i5_feature_code2,...
```

adapter_id

Indicates the DRC index of the Host Ethernet Adapter in hexadecimal.

adapter_type

Indicates whether the virtual SCSI, virtual fibre channel, or serial adapter is a client or server. Valid values are client and server.

addl_vlan_ids

The list of additional IEEE 802.1Q virtual LANs (VLANs) on a virtual Ethernet adapter.

alt_console_slot

The location of the virtual I/O slot that contains the alternate console device for the IBM i logical partition. The default value is none.

alt_restart_device_attached

Whether this IOP has an alternate restart device. Valid values are:

- 0 - not attached
- 1 - attached

alt_restart_device_capable

Whether the IOP is capable of being tagged as the alternate restart device. The default values is 0 (not capable).

alt_restart_device_slot

The location of the virtual I/O slot that contains the alternate restart device for the IBM i logical partition. If the load source slot has a value other than none, this attribute is optional. Valid values are:

- Slot number (for virtual I/O)
- None

auto_io_entitled_mem

Indicates whether the I/O entitled memory of the partition is automatically managed. Valid values follow:

- 0 (Not managed/custom)
- 1 (Automatically managed)

This attribute is only displayed by default if the mem_mode attribute is shared. If the value is dedicated and the attribute is requested, the value is null.

auto_mem_region_size

Indicates the optimal size of the memory region, as automatically calculated by the firmware.

Note: The memory region size applies to the entire system and requires you to reboot the entire system.

backplane_phys_loc

Physical location code of the backplane on which the bus resides.

bus_drc_index

DRC index, in hex, of the bus.

bus_drc_name

DRC name of the bus.

bus_grouping

Indicates whether bus grouping is required. Possible values:

- 0 - not required
- 1 - required

bus_id I/O bus unique ID.**capabilities**

Values currently applied to the system.

config Virtual slot configuration state. Possible values follow:

- empty - no adapter
- ethernet - virtual Ethernet adapter
- fc - virtual fibre channel adapter
- reserved - reserved slot
- scsi - virtual SCSI adapter
- serial - virtual serial adapter
- vasi - virtual asynchronous services interface
- vmc - virtual management channel adapter

config_conn_speed

The configured connection speed of the port in megabits per second. This value can be set by the user using **chhwres**. Possible values are:

- auto
- 10
- 100
- 1000
- 10000

config_duplex

Configured duplex value of the port. This value can be set by the user using **chhwres**. Possible values are:

- auto
- half
- full

config_flow_control

Configured flow control value of the port. This value is used for both the receive and transmit flow control. This attribute can also be set using the **chhwres** command. Possible values are:

- 1 (enabled)
- 0 (disabled)

configurable_pool_proc_units

Total number of configurable processing units in the shared processing pool.

configurable_sys_mem

Total amount, in megabytes, of configurable memory on the managed system.

configurable_sys_proc_units

Total number of configurable processing units on the managed system.

conn_state

Connection state or link state of the physical port. Possible values are:

- 1 (up)
- 0 (down)
- unavailable

cq Total number of completion queues.

console_capable

Whether the IOP is capable of being tagged as the console device. Valid values are:

- 0 - not capable
- 1 - capable

console_slot

The location of the virtual I/O slot that contains the console device for the IBM i logical partition. Valid values are:

- Slot number (for virtual I/O)
- None

curr_avail_pool_proc_units

Current number of configurable processing units in the shared processing pool that are not assigned to partitions.

curr_avail_sys_mem

Current amount, in megabytes, of configurable memory on the managed system that is not assigned to partitions.

curr_avail_sys_proc_units

Current number of configurable processing units on the managed system that are not assigned to partitions.

curr_conn_speed

Speed of the port in Mbps. Possible values are:

- 10
- 100
- 1000
- 10000
- unavailable

curr_duplex

Duplex value of the port. Valid values are:

- half
- full
- unavailable

curr_io_entitled_mem

The current I/O entitled memory of the partition in megabytes. This attribute is only displayed, by default, if the value of the mem_mode attribute is Shared. If the value is Dedicated and the attribute is requested, the value is null.

curr_max_mem
Maximum amount of memory, in megabytes, that can be dynamically assigned to the partition.

curr_max_pool_mem
Maximum amount of physical memory that can be assigned to the memory pool without taking the pool offline. This value is in megabytes.

curr_max_proc_units
Maximum number of processing units that can be dynamically assigned to the partition. This attribute is only valid for partitions using shared processors.

curr_max_procs
Maximum number of processors or virtual processors that can be dynamically assigned to the partition.

curr_max_virtual_slots
Maximum number of virtual slots that can be dynamically configured for the partition.

curr_mem
Current amount of memory, in megabytes, assigned to the partition.

curr_mem_weight
The current shared memory weight of the partition. Possible values are 0 through 255. This attribute is only displayed by default if the mem_mode attribute is set to the shared value. If the value is dedicated and the attribute is requested, the value is null.

curr_min_mem
Minimum amount of memory, in megabytes, that can be dynamically assigned to the partition.

curr_min_proc_units
Minimum number of processing units that can be dynamically assigned to the partition. This attribute is only valid for partitions using shared processors.

curr_min_procs
Minimum number of processors or virtual processors that can be dynamically assigned to the partition.

curr_pool_mem
The amount of physical memory currently assigned to the memory pool in megabytes.

curr_port_group_mcs_value
The current MCS value for the port group.

curr_proc_mode
Indicates whether the partition is using dedicated or shared processors. The mode cannot change dynamically. Valid values:

- ded - dedicated
- shared - shared

curr_proc_units
The current number of processing units assigned to the partition. This attribute is only valid for partitions using shared processors.

curr_procs
The current number of processors or virtual processors assigned to the partition.

curr_shared_proc_pool_id

The shared processor pool that this partition is currently participating in. This attribute is only valid for partitions using shared processors.

curr_sharing_mode

The current sharing mode for a partition. Valid values:

- keep_idle_procs - valid with dedicated processor mode
- share_idle_procs - valid with dedicated processor mode
- share_idle_procs_always - valid with dedicated processor mode
- share_idle_procs_active - valid with dedicated processor mode
- cap - capped mode. valid with shared processor mode
- uncap - uncapped mode. valid with shared processor mode

curr_uncap_weight

The current weighted average of processing priority when in uncapped sharing mode. Valid values are 0 - 255.

deconfig_sys_mem

The amount of memory, in megabytes, on the managed system that has been unconfigured. This includes memory that has been unconfigured by the system due to hardware failure, and memory that has been manually unconfigured.

deconfig_sys_proc_units

The number of processing units on the managed system that have been unconfigured. This includes processing units that have been unconfigured by the system due to hardware failure, and processing units that have been manually unconfigured.

description

A description of the I/O adapter which is in the slot.

device_name

The Virtual I/O Server device used as a paging space device in the memory pool.

direct_console_capable

Whether the IOP is capable of being tagged as the console device for directly attached console. Valid values are:

- 0 - not capable
- 1 - capable

drc_index

The DRC index, in hexadecimal, of the I/O slot.

drc_name

The DRC name of the I/O slot.

feature_codes

The most likely feature code for the I/O adapter. The feature code may be different for each operating system. The **adapter_feature_codes** attribute should be used to distinguish the feature code specific to each operating system.

hsl_pool_id

Whether the logical partition participates in the high-speed Link (HSL) pool. The default value is 0 (does not participate).

ieee_virtual_eth

Indicates whether the virtual Ethernet adapter is using IEEE 802.1Q. Valid values follow:

- 0 - No
- 1 - Yes

- ieq** Total number of interruptible event queues.
- installed_sys_mem**
Total amount, in megabytes, of memory installed on the managed system.
- installed_sys_proc_units**
Total number of processing units installed on the managed system.
- io_pool_id**
A unique identifier for an I/O pool. I/O pools are used in some I/O clustering environments.
- iop** Indicates whether the I/O adapter is an I/O processor. Valid values:
- 0 - no
 - 1 - yes
- iop_info_stale**
Indicates whether the information about the I/O processor is stale. Valid values:
- 0 - no
 - 1 - yes
- is_required**
Indicates whether the I/O slot or virtual I/O adapter is required for the partition.
Valid values:
- 0 - no
 - 1 - yes
- is_trunk**
Indicates whether the virtual Ethernet adapter is the trunk or uplink adapter for the virtual LAN. Valid values:
- 0 - no
 - 1 - yes
- lan_console_capable**
Whether the IOP is capable of being tagged as the console device for a LAN console. The default values is 0 (not capable).
- load_source_attached**
Whether this IOP has a load source device. Valid values:
- 0 - not attached
 - 1 - attached
- load_source_capable**
Whether this IOP is capable of being tagged as the load source device. The default values is 0 (not capable).
- load_source_slot**
The location of the virtual I/O slot that contains the load source for the IBM i logical partition. If the `alt_restart_device_slot` has a value other than none, then this attribute is optional. Valid values are:
- Slot number (for virtual I/O)
 - None

logical_port_id

The ID of the logical port.

logical_port_ids

Logical port IDs defined on the physical port.

lpar_id Partition ID. Valid values are 1 through the maximum number of partitions supported on the managed system (max_lpars).

When using the --subtype pgdev flag, this attribute represents the partition ID of the shared memory partition to which this paging device is associated. If no partition is associated, then the value is none.

lpar_ids

List of logical partition IDs of logical partitions using a pool.

lpar_name

User-defined name for the partition.

When using the --subtype pgdev flag, this attribute represents the name of the shared memory partition to which this paging device is associated. If no partition is associated, this attribute is not displayed by default. If no partition is associated but the attribute is requested, the attribute will have a blank value.

lpar_names

List of names of logical partitions using a pool.

mac_addr

MAC address for the virtual Ethernet adapter.

mac_prefix

The first 3 bytes of the MAC address to be assigned to all virtual Ethernet adapters for this managed system. This must be specified as a 3 byte hexadecimal value (for example, 32ab10) and can only be configured in the **mkgencfg** command.

max_io_entitled_mem_usage

The highest level of I/O entitled memory used by the specified logical partition since the value was last reset.

max_io_pools

Maximum number of I/O pools supported on the managed system.

max_mem_pools

Maximum number of memory pools supported. If the system does not support memory pools, then the value is 0; otherwise, it is 1.

max_paging_vios_per_mem_pool

Maximum number of paging Virtual I/O Server (VIOS) partitions that can be assigned to a memory pool. If the system supports memory pools, this value is 1.

max_procs_per_lpar

Indicates the largest number of dedicated processors that you can assign as the maximum processors value for a logical partition.

max_rcv_packet_size

Maximum Transmission Unit (MTU) size of the port. This value can also be changed using the **chhwres** command. Possible values:

- 1500 (default)
- 9000 (jumbo)

max_shared_proc_pools

Maximum number of shared processing pools which are supported on the managed system.

max_virtual_procs_per_lpar

Indicates the largest number of virtual processors that can be assigned as the maximum processors value for a logical partition.

- max_vlans_per_port**
Indicates the largest number of VLANs that can be specified as additional IEEE 802.1Q VLANs on a virtual Ethernet adapter.
- mem_mode**
Partition memory mode.
- **ded**: Dedicated memory
 - **shared**: Shared memory
- mem_region_size**
The memory region size, in megabytes, for the managed system. The memory regions size represents the granularity of memory allocation to partitions.
- min_io_entitled_mem**
The minimum I/O entitled memory required for the logical partition to function with the current I/O configuration.
- min_proc_units_per_virtual_proc**
Minimum number of processing units that are required for each virtual processor assigned to a partition.
- mr** Total number of memory regions.
- nieq** Total number of non-interruptible event queues.
- num_wwpns_remaining**
The number of worldwide port names that remain to be allocated on the managed system.
- op_console_attached**
Whether this IOP has an operations console device. This attribute is only shown if iop is 1. This attribute is current only if vpd_stale is 0. Valid values are:
- 0 - not capable
 - 1 - attached
- op_console_capable**
Whether this IOP is capable of being tagged as the console device for operations console. This attribute is only shown if iop is 1. This attribute is current only if vpd_stale is 0. The default value is 0 (not capable).
- op_console_slot**
The location of the virtual I/O slot that contains the directly attached Operations Console device for the IBM i logical partition. The default value is none.
- optimal_io_entitled_mem**
The amount of I/O entitled memory that would satisfy the requested configuration of all I/O devices.
- paging_storage_pool**
The name of default paging storage pool. This attribute is the storage pool from which new paging devices are created. If there is no default paging storage pool, the value is an empty string.
- paging_vios_id**
The partition ID of the paging VIOS partition.
- paging_vios_ids**
A comma-separated list of partition IDs for paging VIOS partitions associated with this memory pool. This attribute is not displayed by default. If requested, the value displayed is 1.

paging_vios_name

The name of the paging VIOS partition.

paging_vios_names

A comma-separated list of partition names for paging VIOS partitions that are associated with this memory pool. This attribute is not displayed by default. If requested, the value displayed is the name of the Virtual I/O Server logical partition.

parent_slot_drc_index

The DRC index, in hex, of the parent slot. A value of none indicates there is no parent slot.

pci_class

PCI class code for the I/O adapter. This value is displayed in hexadecimal.

pci_device_id

ID of the device for this I/O adapter. This value is displayed in hexadecimal.

pci_revision_id

Vendor-assigned code indicating the revision number of the I/O adapter. This value is displayed in hexadecimal.

pci_subs_device_id

Vendor-assigned code identifying the type of driver that is installed for the I/O adapter. This value is displayed in hexadecimal.

pci_subs_vendor_id

ID of the vendor that made the driver for the I/O adapter. This value is displayed in hexadecimal.

pci_vendor_id

ID of the vendor that made this I/O adapter. This value is displayed in hexadecimal.

pend_avail_pool_proc_units

After system restart, the number of configurable processing units in the shared processing pool that are not assigned to partitions.

pend_avail_sys_mem

After system restart, the amount, in megabytes, of configurable memory on the managed system that is not assigned to partitions.

pend_avail_sys_proc_units

After system restart, the number of configurable processing units on the managed system that are not assigned to partitions.

pend_io_entitled_mem

The pending I/O entitled memory of the partition in megabytes. This attribute is only displayed by default if the `mem_mode` attribute has a value of `shared`. If the value is `dedicated` and the attribute is requested, the value is `null`.

pend_max_mem

After partition restart, the maximum amount of memory, in megabytes, that can be dynamically assigned to the partition.

pend_max_pool_mem

The pending value for the maximum amount of physical memory that can be assigned to the memory pool without taking the memory pool offline. This value is in megabytes.

pend_max_proc_units

After partition restart, the maximum number of processing units that can be dynamically assigned to the partition. This attribute is only valid for partitions using shared processors.

pend_max_procs

After partition restart, the maximum number of processors or virtual processors that can be dynamically assigned to the partition.

pend_max_virtual_slots

After partition restart, the maximum number of virtual slots that can be dynamically created for the partition.

pend_mem

The target amount of memory, in megabytes, assigned to the partition.

pend_mem_region_size

This is the memory region size of the system after the system is restarted.

pend_mem_weight

The pending shared memory weight of the partition. Possible values are 0 - 255. This attribute is only displayed by default if the mem_mode attribute is set to the shared value. If the value is dedicated and the attribute is requested, the value is null.

pend_min_mem

After partition restart, the minimum amount of memory, in megabytes, that can be dynamically assigned to the partition.

pend_min_proc_units

After partition restart, the minimum number of processing units that can be dynamically assigned to the partition. This attribute is only valid for partitions using shared processors.

pend_min_procs

After partition restart, the minimum number of processors or virtual processors that can be dynamically assigned to the partition.

pend_pool_mem

The amount of physical memory to be assigned to the memory pool in megabytes.

pend_port_group_mcs_value

Pending MCS value of the port group. Setting the MCS value may require a system restart.

pend_proc_mode

Indicates whether the partition will be using dedicated or shared processors after restart. Valid values:

- **ded**: Dedicated
- **shared**: Shared

pend_proc_units

The target number of processing units assigned to the partition. This attribute is only valid for partitions using shared processors.

pend_procs

The target number of processors or virtual processors assigned to the partition.

pend_shared_proc_pool_id

The shared processor pool that a partition is will participate in after restart. This attribute is only valid for partitions using shared processors.

pend_sharing_mode

The target sharing mode for a partition. Valid values:

- keep_idle_procs: Valid with pending dedicated processor mode
- share_idle_procs: Valid with pending dedicated processor mode
- cap - capped mode: Valid with pending shared processor mode
- uncap - uncapped mode: Valid with pending shared processor mode

pend_uncap_weight

The target weighted average of processing priority when in uncapped sharing mode. Valid values are: 0 - 255.

phys_loc

If the device has a physical location code for the slot, this attribute is displayed by default with the physical location code as the value. If the device does not have a physical location code, this attribute is not displayed by default and has a blank value.

phys_port_id

The ID of the physical port in the port group (0 or 1).

phys_port_ids

List of host Ethernet adapter (HEA) physical port IDs in a port group.

phys_port_loc

Location code suffix for the physical port.

phys_port_state

State of the physical port.

phys_port_type

Speed of the port.

port_group

Indicates the Host Ethernet Adapter port group(s).

port_vlan_id

Port virtual LAN ID for the virtual ethernet adapter.

possible_mem_region_size

The memory region size that is possible of the system.

promisc_lpar_id

ID assigned to the promiscuous logical partition. Possible values:

- none
- 1 - 254

promisc_lpar_name

Name assigned to the promiscuous logical partition.

qp

The total number of queue pairs.

recent_alt_restart_device_slot

The previous alternate restart device slot.

rcv_flow_control

Receive flow control value of the port

- 1 (enabled)
- 0 (disabled)
- unavailable

recv_octets, recv_packets_0_64, recv_packets_65_127, recv_packets_128_255, recv_packets_256_511, recv_packets_512_1023, recv_packets_1024_max, recv_packets_dropped_bad_FCS, recv_packets_dropped_internal_mac_error, recv_packets_dropped_in_range_length_error, recv_packets_dropped_out_of_range_length_error, recv_packets_dropped_frame_too_long, recv_packets_dropped_jabber, recv_symbol_error, recv_code_error, recv_runt_frame, recv_fragments, recv_unsupported_opcode, recv_control_pause_frames, recv_search_busy, recv_packets_dropped_filter, recv_packets_dropped_other, recv_packets_dropped_alignment, default_unicast_QPN, recv_MC_packets, recv_BC_packets, trans_MC_packets, trans_BC_packets, trans_octets, trans_packets_length_0_64, trans_packets_length_65_127, trans_packets_length_128_255, trans_packets_length_255_511, trans_packets_length_512_1023, trans_packets_length_1024_max, trans_packets_dropped_bad_FCS, trans_control_pause_frames, trans_tx_local_fault_packets, trans_tx_remote_fault_packets, trans_tx_packets_dropped_int_MAC_error, trans_packets_retried_single_collision, trans_packets_retried_multiple_collision, trans_packets_signal_quality, trans_packets_deferred, trans_packets_late_collisions, trans_packets_excessive_collisions, trans_packets_no_carrier, recv_overrun_frames_dropped

Performance metric values

remote_lpar_id

For client adapters, this specifies the ID of the partition which has the virtual serial/SCSI server adapter for this adapter. For server adapters, this specifies the ID of the partition which has the only client virtual serial/SCSI adapter allowed to connect to this adapter. A value of any indicates that any client virtual serial/SCSI adapter should be allowed to connect to this adapter.

remote_lpar_name

The partition name which corresponds to the remote_lpar_id.

remote_slot_num

For client adapters, this specifies the virtual slot number of the virtual serial/SCSI server adapter for this adapter. For server adapters, this specifies the virtual slot number of the only client virtual serial/SCSI adapter allowed to connect to this adapter. A value of any indicates that any client virtual serial/SCSI adapter should be allowed to connect to this adapter.

required_min_mem_aix_linux

For an AIX or Linux partition, this is the required minimum memory amount, in megabytes, for the maximum memory amount specified with the **--maxmem** parameter.

required_min_mem_os400

The required minimum memory amount, in megabytes, for the maximum memory amount specified with the **--maxmem** parameter. This attribute only applies for an IBM i logical partition.

run_io_entitled_mem

The I/O entitled memory of the partition at run time in megabytes. This attribute is only displayed by default if the mem_mode attribute has a value of shared. If the value is dedicated and the attribute is requested, the value is null.

run_mem

Current amount of memory, in megabytes, that the partition has varied on.

run_mem_weight

The runtime shared memory weight of the logical partition. Possible values are 0 - 255. This attribute is only displayed by default if the mem_mode attribute is set to the shared value. If the value is dedicated and the attribute is requested, the value is null.

run_min_mem

The amount of memory, in megabytes, returned by a running partition's operating system to represent the smallest memory assignment which is currently supported by the operating system. If the operating system does not support this feature, then the value is 0.

run_proc_units

Number of processing units that are varied on for the partition.

run_procs

Number of processors or virtual processors that are varied on for the partition.

run_uncap_weight

The operating system's current setting for the weighted average of processing priority when in uncapped sharing mode. Valid values are: 0 - 255.

shared_proc_pool_id

A unique decimal identifier for a shared processing pool.

size The size of the paging device in megabytes.

slot_io_pool_id

A unique decimal identifier of the I/O pool to which the slot is assigned. A value of none indicates that the slot is not assigned to any I/O pools.

slot_num

Virtual slot number.

slots A list of slot DRC indices.

state The state of a virtual slot. A virtual slot must be able to transition into the off state before a dynamic reconfiguration of the slot can succeed. Valid states follow:

- 0 - The virtual slot is ready for dynamic reconfiguration.
- 1 - The virtual slot is not yet ready for dynamic reconfiguration.

When using paging devices, state indicates the state of the paging device in megabytes. Valid states follow:

- Active - The paging device is in use.
- Inactive - The paging device is available.
- Missing backing device - The backing device being used as a paging space device is missing.
- Defined - The backing device being used as a paging space device is defined.

storage_pool

If the device comes from a storage pool, this attribute is displayed by default with the storage pool as the value. If the device does not come from a storage pool, this attribute is not displayed by default and has a blank value.

supports_hmc

Indicates whether or not the virtual serial adapter is enabled for HMC client support. Valid values:

- 0 - no
- 1 - yes

sys_firmware_mem

The amount of memory, in megabytes, on the managed system that is being used by system firmware.

sys_firmware_pool_mem

The amount of memory in the shared memory pool that is reserved for firmware use. This is the greater of the current and pending values.

trans_flow_control

Transmit flow control value of the port. Possible values are:

- 1 (enabled)
- 0 (disabled)
- unavailable

trunk_priority

The valid values are integers between 1 and 15, inclusive. This value is required for a trunk adapter.

twinax_attached

Whether the IOP has a twinaxial device. This attribute is only shown if iop is 1. This attribute is current only if vpd_stale is 0. Valid values are:

- 0 - not attached
- 1 - attached

twinax_capable

Whether the IOP is capable of being tagged as the console device for a twinaxial console. This attribute is only shown if iop is 1. This attribute is current only if vpd_stale is 0. The default values is 0 (not capable).

type The type of the underlying device. Possible values follow:

- logical - logical device
- phys - physical device

unassigned_logical_port_ids

List of logical port IDs in the port group that are not assigned to a partition.

unit_phys_loc

Physical location code of the I/O unit.

user_def_mac_addr

User defined MAC address for the logical port.

valid_port_group_mcs_values

Valid MCS values.

virtual_opti_pool_id

The location of the virtual I/O slot that contains the directly attached operations console device for the IBM i logical partition. The default value is 0 (does not participate).

vlan_id_list	List of virtual LAN IDs that the logical port has access to.
vpd_model	Model of the I/O adapter.
vpd_serial_num	Serial number of the I/O adapter.
vpd_stale	Indicates whether or not type, model, and serial number of the adapter is not current. Valid values: <ul style="list-style-type: none"> • 0 - not current • 1 - current
vpd_type	Type of the I/O adapter.
wwpns	If the adapter is of type client, the worldwide port names assigned to this adapter are displayed as a comma-separated list. If the adapter is of type server, by default, the attribute is not displayed. If the attribute is requested, the value is null.
wwpn_prefix	A 12-character hexadecimal string that is used by the managed system to generate worldwide port names.
--header	Display a header record, which is a delimiter separated list of attribute names for the attribute values that will be displayed. This header record will be the first record displayed. This option is only valid when used with the -F option.
-R	List information for partitions with resources that can be restored due to a dynamic LPAR operation failure.

Exit Status

This command has a return code of 0 on success.

Security

This command is accessible by all users.

Examples

1. List system level memory information and include the minimum memory required to support a maximum of 1024 MB:

```
lshwres -r mem --level sys --maxmem 1024
```
2. List all memory information for partitions lpar1 and lpar2, and only display attribute values, following a header of attribute names:

```
lshwres -r mem --level lpar --filter "\lpar_names=lpar1,lpar2\" -F --header
```
3. List all I/O units on the system:

```
lshwres -r io --rsubtype unit
```
4. List all virtual Ethernet adapters on the managed system:

```
lshwres -r virtualio --rsubtype eth --level lpar
```
5. List all virtual slots for partition lpar1:

```
lshwres -r virtualio --rsubtype slot --level slot --filter "lpar_names=lpar1"
```
6. List only the installed and configurable processors on the system:

```
lshwres -r proc --level sys -F installed_sys_proc_units,
configurable_sys_proc_units
```
7. List all physical Host Ethernet Adapter ports on the system:

- ```
lshwres -r hea --subtype phys --level port
```
8. List all logical Host Ethernet Adapter ports on the system:

```
lshwres -r hea --subtype logical --level port
```
  9. List the physical Host Ethernet Adapter adapters on the system:

```
lshwres -r hea --subtype phys --level sys
```
  10. List the Host Ethernet Adapter port groups on the system:

```
lshwres -r hea --subtype phys --level port_group
```

## Related Information

The `lssyscfg` and `chhwres` commands.

---

## IVM Isled command

### Purpose

List states for physical and virtual LEDs. This command is valid only in an Integrated Virtualization Manager environment.

### Syntax

To list physical platform LEDs:

```
Isled -r sa -t phys [-F "AttributeNames"] [--header] [-m ManagedSystem]
```

To list virtual partition LEDs:

```
Isled -r sa -t virtuallpar [--filter "FilterData"] [-F "AttributeNames"] [--header] [-m ManagedSystem]
```

To list virtual platform LEDs:

```
Isled -r sa -t virtualsys [-F "AttributeNames"] [--header] [-m ManagedSystem]
```

### Description

The `Isled` command lists the states of physical and virtual LEDs.

### Flags

- |                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-r</b> <i>ResourceType</i>        | The type of LED resources to list. The only valid value is sa for System Attention (SA) LEDs.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>-t</b> <i>SystemAttentionType</i> | The type of System Attention (SA) LEDs to list. <ul style="list-style-type: none"> <li>• <b>-r sa -t virtuallpar</b>: List attributes for virtual partition system attention LEDs <ul style="list-style-type: none"> <li>– Attributes: lpar_id, lpar_name, state</li> <li>– Filters: { lpar_ids   lpar_names }</li> </ul> </li> <li>• <b>-r sa -t virtualsys</b>: List attributes for virtual system attention LEDs <ul style="list-style-type: none"> <li>– Attributes: state</li> <li>– Filters: None</li> </ul> </li> <li>• <b>-r sa -t phys</b>: List attributes for physical system attention LEDs <ul style="list-style-type: none"> <li>– Attributes: state</li> <li>– Filters: None</li> </ul> </li> </ul> |



|                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-m</b> <i>ManagedSystem</i>    | The name of the managed system. This attribute is optional because there is only one system to manage. The name might either be the user-defined name for the managed system, or be in the form <code>ttt-mmm*sssssss</code> , where <code>ttt</code> is the machine type, <code>mmm</code> is the model, and <code>sssssss</code> is the serial number of the managed system.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>--filter</b> <i>FilterData</i> | <p>The filters to apply to the resources to be listed. Filters are used to select which resources of the specified resource type are to be listed. If no filters are used, then all of the resources of the specified resource type will be listed. For example, specific partitions can be listed by using a filter to specify the names or IDs of the partitions to list. Otherwise, if no filter is used, then all the partitions in the managed system will be listed.</p> <p>The filter data consists of filter name/value pairs, which are in comma separated value (CSV) format. The filter data must be enclosed in double quotation marks.</p> <p>The format of the filter data is as follows:<br/> <code>"filter-name=value,filter-name=value,..."</code></p> <p>Note that certain filters accept a comma separated list of values, as follows:<br/> <code>"filter-name=value,value,...",..."</code></p> <p>When a list of values is specified, the filter name/value pair must be enclosed in double quotation marks. Depending on the shell being used, nested double quotation marks might need to be preceded by an escape character, which is usually a <code>'\'</code> character.</p> <p>Unless otherwise indicated, multiple values can be specified for each filter.</p> <p><b>Valid filter names for -r lpar</b></p> <p><b>lpar_ids</b><br/> ID of the partitions to view</p> <p><b>lpar_names</b><br/> Name of the partitions to view</p> |
| <b>-F</b> <i>AttributeNames</i>   | <p>A delimiter separated list of attribute names for the desired attribute values to be displayed for each resource. If no attribute names are specified, then values for all of the attributes for the resource will be displayed.</p> <p>When this option is specified, only attribute values will be displayed. No attribute names will be displayed. The attribute values displayed will be separated by the delimiter which was specified with this option.</p> <p>This option is useful when only attribute values are desired to be displayed, or when the values of only selected attributes are desired to be displayed.</p> <p><b>Attribute Names</b></p> <ul style="list-style-type: none"> <li>• <b>lpar_id</b>: Unique integer identifier for the partition</li> <li>• <b>lpar_name</b>: name of the partition</li> <li>• <b>state</b>: The current state of the LED. Valid values: <ul style="list-style-type: none"> <li>– off - the LED is off</li> <li>– on - the LED is on</li> </ul> </li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>--header</b>                   | Display a header record, which is a delimiter separated list of attribute names for the attribute values that will be displayed. This header record will be the first record displayed. This option is only valid when used with the -F option.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

## Exit Status

This command has a return code of 0 on success.

## Security

This command is accessible by all users.

## Examples

1. Display the physical system attention LED for the system:  

```
lsled -r sa -t phys
```
2. Display all of the virtual partition system attention LEDs:  

```
lsled -r sa -t virtualpar
```
3. Display the virtual partition system attention LEDs for partitions lpar1 and lpar2:  

```
lsled -r sa -t virtualpar --filter \"lpar_names=lpar1,lpar2\"
```

## Related Information

The `chled` command.

---

## lsparinfo command

### Purpose

Displays the logical partition number and name.

### Syntax

`lsparinfo`

### Description

The `lsparinfo` command displays LPAR number and LPAR name. If LPAR does not exist, -1 is displayed for LPAR number and NULL for LPAR name.

### Examples

1. To display the logical partition number and name, type:  

```
lsparinfo
```

---

## IVM lsparmigr command

### Purpose

Lists and displays partition migration information

### Syntax

```
lsparmigr -r manager | lpar | msp | procpool | sys | virtualio | mempool | pgdev [-m <managed system>] [-t <managed system>] [--ip <target HMC/IVM IP address>] [-u <target HMC/IVM username>] [--filter "<filter data>"] [-F [<attribute names>]] [--redundantpgvios { 0 | 1 | 2 }] [--header] [--help]
```

### Description

The `lsparmigr` command displays the state of the partition migration and the adapter mappings that might be used during the partition migration.

## Flags

- r** The type of resources for which to list partition migration information.
- Specify **lpar** to list partition migration information for all of the partitions in managed-system.
- Specify **mosp** to list possible source and destination mover service partition (MSP) pairs for an active partition migration of the partition specified with the **lpar\_names** or **lpar\_ids** filter.
- Specify **procpool** to list possible shared processor pools in the destination managed system that have enough available processing resources for the partition specified with the **lpar\_names** or **lpar\_ids** filter.
- Specify **sys** to list the partition mobility capabilities for managed-system.
- Specify **virtualio** to list possible and suggested mappings of the virtual SCSI adapters and virtual fibre channels in the partition specified with the **lpar\_names** or **lpar\_ids** filter to Virtual I/O Server in the destination managed system.
- Specify **mempool** to list details for each shared memory pool on the target managed system.
- Specify **pgdev** to list paging devices available on the target managed system.
- m** *<managed system>* The name of the source managed system.
- t** *<managed system>* The name of target managed system.
- ip** The IP address or hostname of the Integrated Virtualization Manager managing the target system.
- u** The user name to use on the Integrated Virtualization Manager that manages the target system. If the **--ip** flag is specified and **-u** flag is not specified, the user name on the source Integrated Virtualization Manager is used on the target Integrated Virtualization Manager.
- filter** *<filter data>* The filter or filters to apply to the resources to be listed. Filters are used to select which resources are to be listed. If no filters are used, then all of the resources will be listed. For example, partition information can be listed by using a filter to specify the names or IDs of the partitions to list. Otherwise, if no filter is used, then information for all of the partitions in the managed system will be listed.
- The filter data consists of filter name/value pairs, which are in comma separated value (CSV) format. The filter data must be enclosed in quotation marks.
- The format of the filter data is as follows:  
"filter-name=value,filter-name=value,..."
- Note:** Certain filters accept a comma separated list of values, as follows:  
"filter-name=value,value,...",..."
- When a list of values is specified, the filter name/value pair must be enclosed in quotation marks. Depending on the shell being used, nested quotation marks may need to be preceded by an escape character, which is usually a `'\'` character.
- Valid filter names:  
lpar\_names | lpar\_ids  
Only one of these filters may be specified
- F** [*<attribute names>*] List of the names of the attributes. If no attribute names are specified, then all attributes will be listed.

|                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-redundantpgvios</b> | Specify the configured level of redundancy for the memory pools to be listed.<br><b>0</b> Do not require redundancy on the target managed system.<br><b>1</b> Only return memory pool information for those that are capable of redundancy.<br><b>2</b> Return all memory pool information regardless of redundancy capabilities.<br>If you do not specify the parameter, memory pools capable of the currently configured level of redundancy are returned.<br><br>Exactly one partition must be specified by either the <code>lpar_ids</code> filter or <code>lpar_names</code> filter.<br><br>If no suitable memory pools are found on the destination managed system, the destination IVM generates an error message that describes the problem. |
| <b>--header</b>         | Displays a header record, which is a delimiter separated list of attribute names for the attribute values that are displayed. This header record is the first record displayed. This option is only valid when used with the <code>-F</code> option.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>--help</b>           | Displays the help text for this command and exit.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

## Exit Status

This command has a return code of 0 on success.

## Examples

1. To return partition mobility capabilities for the HMC or the Integrated Virtualization Manager, type the following command:

```
lslparmigr -r manager
remote_lpar_mobility_capable=1
```

2. To return partition mobility capabilities for the specified managed system, type the following command:

```
lslparmigr -m zd25ae -r sys
inactive_lpar_mobility_capable=1,num_inactive_migrations_supported=40,
num_inactive_migrations_in_progress=0,
active_lpar_mobility_capable=1,num_active_migrations_supported=40,
num_active_migrations_in_progress=0
```

3. To return a list of partitions on the specified managed system and their migration state, type the following command:

```
$ lslparmigr -r lpar
name=10-0EDDA,lpar_id=1,migration_state=Not Migrating
name=mptest_migration,lpar_id=2,migration_state=Not Migrating
name=lp3,lpar_id=3,migration_state=Not Migrating
name=mp_i51,lpar_id=4,migration_state=Not Migrating
name=mover2,lpar_id=5,migration_state=Migration Starting,
migration_type=active,dest_sys_name=migfspL4,
dest_lpar_id=5,source_msp_name=10-0EDDA,source_msp_id=1,
dest_msp_name=10-0EDEA,dest_msp_id=1,bytes_transmitted=10247078,
bytes_remaining=259547136,
remote_manager=miglp9.yourcity.yourcompany.com,
remote_user=padmin
name=lpar10,lpar_id=10,migration_state=Not Migrating
```

4. To return a list of possible source and destination MSP pairs, type the following command:

```
lslparmigr -r msp -m migfspL2 -t migfspL1
source_msp_name=VIOS,source_msp_id=2,"dest_msp_names=VIOS,VIOS2",
"dest_msp_ids=1,3","ipaddr_mappings=192.168.10.10/en1/1/VIOS/192.
168.10.11/en1,192.168.20.10/en2/3/VIOS2/192.168.20.13/en0"
source_msp_name=VIOS2,source_msp_id=1,"dest_msp_names=VIOS,VIOS2",
"dest_msp_ids=1,3","ipaddr_mappings=192.168.10.12/en0/1/VIOS/192.
168.10.11/en1,192.168.20.12/en1/3/VIOS2/192.168.20.13/en0"
```

5. To return a list of possible and suggested mappings for virtual SCSI and virtual Ethernet adapters, type the following command:

```
lslparmigr -r virtualio -m migfspL2 -t migfspL1 --filter "lpar_ids=5"
"possible_virtual_scsi_mappings=2/VIOS/1,2/VIOS2/3",
"suggested_virtual_scsi_mappings=2/VIOS/1"
```

6. To return a list of existing shared processor pools on the target managed system, type the following command:

```
lslparmigr -r procpool -m migfspL2 -t migfspL1 --filter "lpar_ids=5"
"shared_proc_pool_ids=0,3,6","shared_proc_pool_names=
DefaultPool,ProcPoolA,ProcPoolB"
```

7. To return a list of possible and suggested mappings for virtual fibre channel adapters, type the following command:

```
lslparmigr -r virtualio -m migfspL2 -t migfspL1 --filter "lpar_ids=5"
"possible_virtual_scsi_mappings=2/VIOS/1",
"suggested_virtual_scsi_mappings=2/VIOS/1"
"possible_virtual_fc_mappings=4/VIOS/1,4/VIOS2/2",
"suggested_virtual_fc_mappings=4/VIOS/1"
```

---

## IVM lslparutil command

### Purpose

List utilization metrics for the managed system and partitions. This command is available only in an Integrated Virtualization Manager environment.

### Syntax

To list utilization metrics:

```
lslparutil {-r sys | pool | lpar | mempool} [--startyear Year] [--startmonth Month] [--startday Day] [
--starthour Hour] [--startminute Minute] [--endyear Year] [--endmonth Month] [--endday Day] [
--endhour Hour] [--endminute Minute] [-n NumberOfEvents] [-spread] [--filter "FilterData"] [-F
"AttributeNames"] [--header] [-m ManagedSystem]
```

To list utilization configuration attributes:

```
lslparutil -r config [-F "AttributeNames"] [--header] [-m ManagedSystem]
```

### Description

The **lslparutil** command lists utilization metrics for the managed system and partitions. The sample rate at which these metrics are gathered can be changed, or disabled. The information gathered from this command allows for the determination of processor utilization on a system and partition level over a given period of time.

## Flags

**-r** *ResourceType*

The type of resources to list:

- **-r config**: List configuration attributes for utilization monitoring
  - **Attributes**: sample\_rate
  - **Filters**: None
- **-r lpar**: List utilization data for logical partitions
  - **Attributes**: time, time\_bin, sys\_time, event\_type, resource\_type, time\_cycles, lpar\_id, uptime, curr\_io\_entitled\_mem, curr\_proc\_mode, curr\_proc\_units, curr\_procs, curr\_sharing\_mode, curr\_uncap\_weight, curr\_5250\_cpw\_percent, curr\_mem, entitled\_cycles, capped\_cycles, mapped\_io\_entitled\_mem, mem\_mode, mem\_overage\_cooperation, phys\_run\_mem, uncapped\_cycles, donated\_cycles, idle\_cycles, run\_latch\_instructions, run\_latch\_cycles, run\_mem\_weight
  - **Filters**: { lpar\_ids | lpar\_names}
- **-r mempool**: List information and statistics about memory pools
  - **Attributes**: time, time\_bin, sys\_time, event\_type, resource\_type, page\_faults, page\_in\_delay, curr\_pool\_mem, lpar\_curr\_io\_entitled\_mem, lpar\_mapped\_io\_entitled\_mem, lpar\_run\_mem, sys\_firmware\_pool\_mem
- **-r pool**: List utilization data for shared processor pools
  - **Attributes**: time, time\_bin, sys\_time, event\_type, resource\_type, time\_cycles, shared\_proc\_pool\_id, curr\_avail\_pool\_proc\_units, configurable\_pool\_proc\_units, borrowed\_pool\_proc\_units, total\_pool\_cycles, utilized\_pool\_cycles
  - **Filters**: None
- **-r sys**: List utilization data for the managed system
  - **Attributes**: time, time\_bin, sys\_time, event\_type, resource\_type, state, configurable\_sys\_proc\_units, configurable\_sys\_mem, curr\_avail\_sys\_proc\_units, curr\_avail\_5250\_cpw\_percent, curr\_avail\_sys\_mem, sys\_firmware\_mem, proc\_cycles\_per\_second
  - **Filters**: None

**--startyear** *Year*

Starting year filter (default 1970)

**--startmonth** *Month*

Starting month filter (default 1) January is 1. December is 12

**--startday** *Day*

Starting day filter (default 1)

**--starthour** *Hour*

Starting hour filter (default 0)

**--startminute** *Minute*

Starting minute filter (default 0)

**--endyear** *Year*

Ending year filter (default now)

**--endmonth** *Month*

Ending month filter (default now)

**--endday** *Day*

Ending day filter (default now)

**--endhour** *Hour*

Ending hour filter (default now)

**--endminute** *Minute*

Ending minute filter (default now)

**--n** *NumberOfEvents*

Starting from the most recent event, the maximum number of samples to return.

**--spread**

Used with -n to distribute displayed samples evenly over time.

**-m** *ManagedSystem*

The name of the managed system. This attribute is optional because there is only one system to manage. The name might either be the user-defined name for the managed system, or be in the form tttt-mmm\*sssssss, where tttt is the machine type, mmm is the model, and sssssss is the serial number of the managed system.

**--filter** *FilterData*

The filters to apply to the resources to be listed. Filters are used to select which resources of the specified resource type are to be listed. If no filters are used, then all of the resources of the specified resource type will be listed. For example, specific partitions can be listed by using a filter to specify the names or IDs of the partitions to list. Otherwise, if no filter is used, then all the partitions in the managed system will be listed.

The filter data consists of filter name/value pairs, which are in comma separated value (CSV) format. The filter data must be enclosed in double quotation marks.

The format of the filter data is as follows:

```
"filter-name=value,filter-name=value,..."
```

Note that certain filters accept a comma separated list of values, as follows:

```
"filter-name=value,value,...",..."
```

When a list of values is specified, the filter name/value pair must be enclosed in double quotation marks. Depending on the shell being used, nested double quotation marks might need to be preceded by an escape character, which is usually a backslash character.

Unless otherwise indicated, multiple values can be specified for each filter.

#### **Valid filter names for -r lpar**

##### **lpar\_ids**

ID of the partitions to view

##### **lpar\_names**

Name of the partitions to view

**-F** *AttributeNames*

A delimiter separated list of attribute names for the desired attribute values to be displayed for each resource. If no attribute names are specified, then values for all of the attributes for the resource will be displayed.

When this option is specified, only attribute values will be displayed. No attribute names will be displayed. The attribute values displayed will be separated by the delimiter which was specified with this option.

This option is useful when only attribute values are desired to be displayed, or when the values of only selected attributes are desired to be displayed.

#### **Attribute Names:**

##### **borrowed\_pool\_proc\_units**

Processing units that are being borrowed from powered off partitions with dedicated processors.

##### **capped\_cycles**

The number of capped processing cycles utilized by this partition since the system started.

##### **configurable\_pool\_proc\_units**

The number of configurable processing units in the shared pool at the time of the sample

##### **configurable\_sys\_mem**

The amount of configurable system memory (in megabytes) at the time of the sample.

**configurable\_sys\_proc\_units**

The number of configurable system processing units at the time of the sample.

**curr\_5250\_cpw\_percent**

The 5250 CPW percent assigned to the partition at the time of the sample.

**curr\_avail\_5250\_cpw\_percent**

The 5250 CPW percent available to be assigned to partitions at the time of the sample.

**curr\_avail\_pool\_proc\_units**

The number of processing units available to be assigned to partitions at the time of the sample.

**curr\_avail\_sys\_mem**

The amount of memory (in megabytes) available to be assigned to partitions at the time of the sample.

**curr\_avail\_sys\_proc\_units**

The number of processing units available to be assigned to partitions at the time of the sample.

**curr\_io\_entitled\_mem**

The size of memory that the logical partition is entitled to map to I/O (in megabytes).

**curr\_mem**

The amount of memory (in megabytes) assigned to the partition at the time of the sample.

**curr\_pool\_mem**

The total amount of physical memory (in megabytes) in the memory pool.

**curr\_proc\_mode**

The processing mode for the partition at the time of the sample. Valid values

- ded: dedicated processor mode
- shared: shared processor mode curr\_proc\_units

**curr\_proc\_units**

Current number of processing units assigned to the partition. This attribute is only valid for partitions using shared processors.

**curr\_procs**

The number of processors or virtual processors assigned to the partition at the time of the sample.

**curr\_sharing\_mode**

The sharing mode of the partition at the time of the sample. Valid values:

- keep\_idle\_procs: valid with dedicated processor mode
- share\_idle\_procs: valid with dedicated processor mode
- cap: capped mode. valid with shared processor mode
- uncap: uncapped mode. valid with shared processor mode

**curr\_uncap\_weight**

The current weighted average of processing priority when in uncapped sharing mode at the time of the sample. The smaller the value, the lower the weight. Valid values are 0-255

**donated\_cycles**

The number of cycles donated by a partition with dedicated processors.



**entitled\_cycles**

The number of processing cycles to which the partition has been entitled since the system started. This value is based on the number of processing units assigned to the partition, and might be greater than, or smaller than the number of cycles actually used.

**event\_type**

The type of event. This will be displayed with a constant value of sample for all samples except the config resource type.

**idle\_cycles**

The number of cycles that the logical partition reported as idle. If the partition does not have the ability to report idle cycles, this value is 0.

**lpar\_curr\_io\_entitled\_mem**

The total entitlement (in megabytes) of all shared memory partitions served by the memory pool.

**lpar\_id** The unique integer identifier for the partition.

**lpar\_mapped\_io\_entitled\_mem**

The total I/O mapped memory (in megabytes) of all active partitions served by the memory pool.

**lpar\_run\_mem**

The total logical real memory in megabytes of all active partitions served by the memory pool.

**mapped\_io\_entitled\_mem**

Size of memory (in megabytes) that the logical partition has mapped to I/O memory.

**mem\_mode**

The partition memory mode. Valid values:

- ded - dedicated memory
- shared - shared memory

If the memory mode is shared, then the logical partition cannot be assigned any physical I/O slots, the `proc_mode` attribute on the partition must be shared, and a memory pool must exist.

**Note:** The Virtual I/O Server logical partition only supports the dedicated memory mode.

**page\_faults**

The total-page fault count of the memory pool since the initialization of the pool, or since the last reboot of the hypervisor, whichever is more recent.

**page\_in\_delay**

The total page in delay (in microseconds) spent waiting for the page faults since the initialization of the memory pool, or since the last reboot of the hypervisor, whichever is more recent.

**phys\_run\_mem**

The size of physical memory (in megabytes) that is backing the logical address space of the logical partition. This value represents the portion of the logical memory assignment that is currently paged in the logical partition.

**proc\_cycles\_per\_second**

The processing cycles per second on one physical processor. This value is static for a particular managed system.

**resource\_type**

The resource type queried. Valid values are sys, pool, and lpar, depending on which value is supplied for the **-r** flag.

**run\_latch\_cycles**

The number of nonidle cycles used by the partition, while the run-latch is set, since the last time the system was started.

**run\_latch\_instructions**

The number of nonidle instructions performed by the partition, while the run-latch is set, since the last time the system was started.

**run\_mem\_weight**

Shared memory weight at run time.

**sample\_rate**

The rate at which samples are obtained. This can be changed with the **chlpoutil** command. Valid values:

- 0: Samples will not be retrieved.
- 30: Samples will be retrieved every 30 seconds. This is the default value.
- 60: Samples will be retrieved every minute.
- 300: Samples will be retrieved every 5 minutes.
- 1800: Samples will be retrieved every 30 minutes.
- 3600: Samples will be retrieved every hour.

**shared\_proc\_pool\_id**

The unique decimal identifier for a shared processing pool.

**state** The state of the managed system at the time of the sample.

**sys\_firmware\_mem**

The amount of memory, in megabytes, on the managed system that is being used by system firmware at the time of the sample.

**sys\_firmware\_pool\_mem**

The amount of memory in the shared memory pool that is reserved for use by the firmware. This is the greater of the current and pending values.

**sys\_time**

The time on the managed system that the sample was taken. The time and sys\_time attributes will have the same value.

**time** The time on the management partition that this sample was taken. The time and sys\_time attributes will have the same value.

**time\_bin**

The time or sys\_time represented as the number of milliseconds since January 1, 1970.

**time\_cycles**

The number of time cycles since the system was started.

**total\_pool\_cycles**

The total number of processing cycles available in the shared pool since the system was started.

**uncapped\_cycles**  
The number of uncapped processing cycles utilized by this partition since the system was started.

**uptime** The amount of time (in seconds) that the partition has been running at the time of the sample.

**utilized\_pool\_cycles**  
The number of processing cycles in the shared pool that have been utilized since the system was started.

**--header** Display a header record, which is a delimiter separated list of attribute names for the attribute values that will be displayed. This header record will be the first record displayed. This option is only valid when used with the -F option.

## Exit Status

This command has a return code of 0 on success.

## Security

This command is accessible by all users.

## Examples

1. To list the last five shared processing pool utilization metrics, type:

```
lslparutil -r pool -n 5
```

2. To calculate the shared processing pool utilization in percent over a five-minute time period, type:

```
lslparutil -r pool --startyear 2006 --startmonth 10 --startday 4
--starthour 15 --startminute 23 --endyear 2006 --endmonth 10
--endday 4 --endhour 15 --endminute 28
-F time,total_pool_cycles,utilized_pool_cycles
10/04/2006 15:27:56,449504263518104,372389272879
10/04/2006 15:27:26,449442382657200,372195239995
10/04/2006 15:26:56,449382561908822,371998920942
10/04/2006 15:26:26,449320667371346,371797239591
10/04/2006 15:25:56,449258781703684,371592366752
10/04/2006 15:25:26,449196894951060,371394157970
10/04/2006 15:24:56,449135006535822,371192089089
10/04/2006 15:24:27,449075176369863,369243635687
10/04/2006 15:23:56,449013298087726,369040980263
10/04/2006 15:23:26,448951445376558,368850062933

Pool utilization = (utilized_pool_cycle / total_pool_cycles) * 100
Pool utilization = ((372389272879 - 368850062933) /
(449504263518104 - 448951445376558)) * 100
Pool utilization = 0.64%
```

3. To calculate the shared processing pool size over a five-minute time period, type:

```
lslparutil -r pool --startyear 2006 --startmonth 10 --startday 4
--starthour 15 --startminute 23 --endyear 2006 --endmonth 10
--endday 4 --endhour 15 --endminute 28 -F time,time_cycles,total_pool_cycles
10/04/2006 15:27:56,112376065883848,449504263518104
10/04/2006 15:27:26,112360595668767,449442382657200
10/04/2006 15:26:56,112345640481652,449382561908822
10/04/2006 15:26:26,112330166847247,449320667371346
10/04/2006 15:25:56,112314695430447,449258781703684
10/04/2006 15:25:26,112299223741951,449196894951060
10/04/2006 15:24:56,112283751639775,449135006535822
10/04/2006 15:24:27,112268794096846,449075176369863
10/04/2006 15:23:56,112253324526335,449013298087726
10/04/2006 15:23:26,112237861348574,448951445376558
```

```

Pool size = total_pool_cycles / time_cycles
Pool size = (449504263518104 - 448951445376558) /
 (112376065883848 - 112237861348574)
Pool size = 4

```

- To calculate the processing utilization in percent for partition 1 over the last 11 samples, type:

```

lslparutil -r lpar -F time,lpar_id,entitled_cycles,capped_cycles,uncapped_cycles
--filter lpar_ids=1 -n 11

```

```

06/26/2005 12:13:04,1,13487973395246,353524992184,93964052971
06/26/2005 12:12:33,1,13486720703117,353490258336,93964052971
06/26/2005 12:12:03,1,13485467110700,353456792591,93964052971
06/26/2005 12:11:33,1,13484213859686,353423048854,93964052971
06/26/2005 12:11:03,1,13482961098044,353386674795,93964052971
06/26/2005 12:10:32,1,13481706673802,353350985013,93964052971
06/26/2005 12:10:02,1,13480453156357,353317211748,93964052971
06/26/2005 12:09:32,1,13479199972343,353283141535,93964052971
06/26/2005 12:09:02,1,13477946765207,353248812551,93964052971
06/26/2005 12:08:31,1,13476693184663,353213970760,93964052971
06/26/2005 12:08:01,1,13475439617080,353179654833,93964052971

```

```

Processor utilization = ((capped_cycles + uncapped_cycles)
 / entitled_cycles) * 100
Processor utilization = (((353524992184 - 353179654833)
 + (93964052971 - 93964052971))
 / (13487973395246 - 13475439617080)) * 100
Processor utilization = 2.76%

```

- To calculate the most recent number of utilized processor units for partition 1, type:

```

lslparutil -r lpar -F time,time_cycles,capped_cycles,uncapped_cycles -n 2 \
--filter lpar_ids=1

```

```

10/06/2006 09:42:58,190122585897822,836322334068,1209599213218
10/06/2006 09:42:28,190107628555119,836215824328,1209507899652

```

```

Processor units used = (capped_cycles + uncapped_cycles) / time_cycles
Processor units used = ((836322334068 - 836215824328)
 + (1209599213218 - 1209507899652)) /
 (190122585897822 - 190107628555119)
Processor units used = 0.01

```

- To display 5 entries spread over the last day, type:

```

lslparutil -r lpar --startyear 2005 --startmonth 6 --startday 25 \
--starthour 14 --startminute 20 --endyear 2005 --endmonth 6 \
--endday 26 --endhour 14 --endminute 20 -F time,lpar_id,entitled_cycles,\
capped_cycles,uncapped_cycles -n 5 --spread

```

```

06/26/2005 14:20:03,1,64074629124428,2634420796918,3335839807455
06/26/2005 08:20:03,1,63180190141506,2597103712238,3292339240560
06/26/2005 02:20:02,1,62285686629911,2559847748332,3249081303922
06/25/2005 20:20:01,1,61391161857754,2522594738730,3205850397796
06/25/2005 14:20:01,1,60496719757782,2485459490629,3162821474641

```

**Note:** This data is particularly useful for graphing utilization data.

- To view the number of donated cycles:

```

lslparutil -r lpar -F donated_cycles

```

## Related Information

The `chlparutil` and `lssyscfg` commands.

---

## lslv command

### Purpose

Displays information about a logical volume.

## Syntax

`lslv [ -map | -pv ] LogicalVolume [ -field FieldName ] [ -fmt Delimiter ]`

`lslv -free [ -field Fieldname ] [ -fmt Delimiter ]`

## Description

The `lslv` command displays the characteristics and status of the *LogicalVolume* or lists the logical volume allocation map for the physical partitions on the *PhysicalVolume* in which the logical volume is located. The logical volume can be a name or identifier.

If no flags are specified, the following status is displayed:

|                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Logical volume</b>            | Name of the logical volume. Logical volume names must be unique systemwide and can range from 1 to 15 characters.                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Volume group</b>              | Name of the volume group. Volume group names must be unique systemwide and can range from 1 to 15 characters.                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>Logical volume identifier</b> | Identifier of the logical volume.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Permission</b>                | Access permission; read-only or read-write.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Volume group state</b>        | State of the volume group. If the volume group is activated with the <b>activatevg</b> command, the state is either active/complete (indicating all physical volumes are active) or active/partial (indicating all physical volumes are not active). If the volume group is not activated with the <b>activatevg</b> command, the state is inactive.                                                                                                                                                                             |
| <b>Logical volume state</b>      | State of the logical volume. The Opened/stale status indicates the logical volume is open but contains physical partitions that are not current. Opened/syncd indicates the logical volume is open and synchronized. Closed indicates the logical volume has not been opened.                                                                                                                                                                                                                                                    |
| <b>Type</b>                      | Logical volume type.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Write verify</b>              | Write verify state of On or Off.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Mirror write consistency</b>  | Mirror write consistency state of Yes or No.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Max LPs</b>                   | Maximum number of logical partitions the logical volume can hold.                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>PP size</b>                   | Size of each physical partition.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Copies</b>                    | Number of physical partitions created for each logical partition when allocating.                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Schedule policy</b>           | Sequential or parallel scheduling policy.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>LPs</b>                       | Number of logical partitions currently in the logical volume.                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>PPs</b>                       | Number of physical partitions currently in the logical volume.                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>Stale partitions</b>          | Number of physical partitions in the logical volume that are not current.                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>Bad blocks</b>                | Bad block relocation policy.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Inter-policy</b>              | Inter-physical allocation policy.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Strictness</b>                | Current state of allocation. Possible values are strict, nonstrict, or superstrict. A strict allocation states that no copies for a logical partition are allocated on the same physical volume. If the allocation does not follow the strict criteria, is called nonstrict. A nonstrict allocation states that at least one occurrence of two physical partitions belong to the same logical partition. A superstrict allocation states that no partition from one mirror copy may reside the same disk as another mirror copy. |
| <b>Intra-policy</b>              | Intra-physical allocation policy.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Upper bound</b>               | If the logical volume is super strict, upper bound is the maximum number of disks in a mirror copy.                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>Relocatable</b>               | Indicates whether the partitions can be relocated if a reorganization of partition allocation takes place.                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Mount point</b>               | File system mount point for the logical volume, if applicable.                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>Label</b>                     | Specifies the label field for the logical volume.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>PV distribution</b>           | The distribution of the logical volume within the volume group. The physical volumes used, the number of logical partitions on each physical volume, and the number of physical partitions on each physical volume are shown.                                                                                                                                                                                                                                                                                                    |
| <b>striping width</b>            | The number of physical volumes being striped across.                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

**strip size** The number of bytes per stripe.

The **-free** flag displays logical volumes that are available to be used as backing devices for virtual storage.

Full scripting support is available by using the **-field** *FieldNames* and **-fmt** *Delimiter* flags. The **-field** flag will allow the user to select which output fields to display and in what order, while the **-fmt** flag provides scriptable output. The output fields will be displayed in the order they appear on the command line.

## Flags

**-free** Lists only logical volumes that are available for use as a backing device for virtual SCSI.

**-field** Specifies the list of fields to display. The following fields are supported if no flags are specified:

- lvname** Name of the logical volume. Logical volume names must be unique systemwide and can range from 1 to 15 characters.
- vgname** Name of the volume group. Volume group names must be unique systemwide and can range from 1 to 15 characters.
- lvid** Identifier of the logical volume.
- access** Access permission: read-only or read-write.
- vgstate** State of the volume group. If the volume group is activated with the **activatevg** command, the state is either *active/complete* (indicating all physical volumes are active) or *active/partial* (indicating all physical volumes are not active). If the volume group is not activated with the **deactivatevg** command, the state is *inactive*.
- lvstate** State of the logical volume. The *opened/stale* status indicates the logical volume is open but contains physical partitions that are not current. *Opened/syncd* indicates the logical volume is open and synchronized. *Closed* indicates the logical volume has not been opened.
- type** Logical volume type.
- wverify** Write verify state of on or off.
- mwc** Mirror write consistency state of on or off.
- maxlps** Maximum number of logical partitions the logical volume can hold.
- ppsize** Size of each physical partition.
- copies** Number of physical partitions created for each logical partition when allocating.
- spolicy** Sequential or parallel scheduling policy.
- lps** Number of logical partitions currently in the logical volume.
- pps** Number of physical partitions currently in the logical volume.
- stale** Number of physical partitions in the logical volume that are not current.
- bbpolicy** Bad block relocation policy.

**inter** Inter-physical allocation policy.

**intra** Intra-physical allocation policy.

**ubound**

If the logical volume is super strict, upper bound is the maximum number of disks in a mirror copy.

**relocatable**

Indicates whether the partitions can be relocated if a reorganization of partition allocation takes place.

**mount** File system mount point for the logical volume, if applicable.

**label** Specifies the label field for the logical volume.

**separatepv**

The strictness value. Current state of allocation, strict, nonstrict, or superstrict. A strict allocation states that no copies for a logical partition are allocated on the same physical volume. If the allocation does not follow the strict criteria, it is called nonstrict. A nonstrict allocation states that at least one occurrence of two physical partitions belong to the same logical partition. A superstrict allocation states that no partition from one mirror copy may reside the same disk as another mirror copy.

**serialio** Serialization of overlapping IOs state of yes or no. If serialization is turned on (yes), then overlapping IOs are not allowed on a block range, and only a single IO in a block range is processed at any one time. Most applications, such as file systems and databases, perform serialization; therefore, serialization should be turned off (no). The default setting for new logical volumes is no.

The following fields are supported if the **-pv** flag is specified:

**pvname**

Physical volume disk name

**copies** The following three fields:

- The number of logical partitions containing at least one physical partition (no copies) on the physical volume
- The number of logical partitions containing at least two physical partitions (one copy) on the physical volume
- The number of logical partitions containing three physical partitions (two copies) on the physical volume

**inband**

The percentage of physical partitions on the physical volume that belong to the logical volume and were allocated within the physical volume region specified by Intra-physical allocation policy.

**dist** The number of physical partitions allocated within each section of the physical volume: outer edge, outer middle, center, inner middle, and inner edge of the physical volume.

The following fields are supported if the **-map** flag is specified:

**lpnum** Logical partition number.

**pvname1**

Physical volume name where the logical partition's first physical partition is located.

**ppnum1**

First physical partition number allocated to the logical partition.

**pvname2**

Physical volume name where the logical partition's second physical partition (first copy) is located.

**ppnum2**

Second physical partition number allocated to the logical partition.

The following fields are supported if the **-free** flag is specified:

**lvname**

Logical partition number.

**size**

Physical volume name where the logical partition's first physical partition is located.

**vgname**

Name of the volume group. Volume group names must be unique systemwide and can range from 1 to 15 characters.

**-fmt**  
**-map**

Specifies a delimiter character to separate output fields.

Lists the following fields for each logical partition:

**LPs** Logical partition number.

**PV1**

Physical volume name where the logical partition's first physical partition is located.

**PP1**

First physical partition number allocated to the logical partition.

**PV2**

Physical volume name where the logical partition's second physical partition (first copy) is located.

**PP2**

Second physical partition number allocated to the logical partition.

**-pv**

Lists the following fields for each physical volume in the logical volume:

**PV**

Physical volume name.

**Copies** The following three fields:

- The number of logical partitions containing at least one physical partition (no copies) on the physical volume
- The number of logical partitions containing at least two physical partitions (one copy) on the physical volume
- The number of logical partitions containing three physical partitions (two copies) on the physical volume

**In band**

The percentage of physical partitions on the physical volume that belong to the logical volume and were allocated within the physical volume region specified by Intra-physical allocation policy.

**Distribution**

The number of physical partitions allocated within each section of the physical volume: outer edge, outer middle, center, inner middle, and inner edge of the physical volume.



## Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

## Examples

1. To display information about logical volume **lv03**, type:

```
lslv lv03
```

Information about logical volume **lv03**, its logical and physical partitions, and the volume group to which it belongs is displayed.

2. To display information about logical volume **lv03** by physical volume, type:

```
lslv -pv lv03
```

The characteristics and status of **lv03** are displayed, with the output arranged by physical volume.

3. To display a list of logical volumes that can be used as backing devices, type:

```
lslv -free
```

The system displays a message similar to the following:

| LV NAME | SIZE(megabytes) | VOLUME GROUP |
|---------|-----------------|--------------|
| lv00    | 64              | rootvg       |
| lv01    | 64              | rootvg       |

4. To display only the type and volume group of logical volume **hd6** and separate the data by a : (colon) , type:

```
lslv hd6 -field type vgroup -fmt :
```

The system displays a message similar to the following:

```
paging:rootvg
```

## Related Information

The **mklv** command, the **extendlv** command, and the **rmlv** command.

---

## lsmmap command

### Purpose

Displays the mapping between physical, logical, and virtual devices.

### Syntax

```
lsmmap { -vadapter ServerVirtualAdapter | -plc PhysicalLocationCode | -all } [-type BackingDeviceType | -net | -npiv] [-field FieldNames] [-fmt Delimiter]
```

```
lsmmap -ams { -vtd PagingDevice | -all } [-type BackingDeviceType] [-field FieldNames] [-fmt Delimiter]
```

### Description

The **lsmmap** command displays the mapping between virtual host adapters and the physical devices they are backed to. Given an adapter name (*ServerVirtualAdapter*) or physical location code (*PhysicalLocationCode*) of a server virtual adapter, the device name of each connected virtual target device (child devices), its logical unit number, backing device(s) and the backing devices physical location code is displayed. If the **-net** flag is specified the supplied device must be a virtual server Ethernet adapter.

The **-fmt** flag divides the output by a user-specified delimiter/character (delimiter). The delimiter can be any non-white space character. This format is provided to facilitate scripting.

The **-type** flag limits the types of backing devices displayed to the user-specified list, *BackingDeviceType*. With the exception of the net type, any combination of device types may be specified. The net type cannot be combined with any other type.

The **-npiv** flag is used to display the server binding information between the virtual fibre channel and the physical fibre channel adapter. It is also used to display client adapter attributes that are being sent to the server adapter.

## Flags

- all** Specifies the **lsmmap** output that should be displayed for all virtual SCSI devices. If used with the **-net** flag, virtual Ethernet adapters are displayed. If used with the **-npiv** flag, virtual fibre channel adapters are displayed.
- ams** Displays paging space device information used in active memory sharing.
- field *FieldName*** Specifies the list of fields to display.  
**Note:** The order that you specify the attributes is not guaranteed to be the same order as they are returned.

For virtual SCSI (default), the following fields are supported:

- svsa** Server virtual adapter.
- physloc** The physical location code of the server virtual adapter.
- clientid** Client partition ID.
- vtd** Virtual target device.
- lun** Logical unit number.
- backing** Backing device.
- bdphysloc** The physical location code of the backing device.
- status** Virtual target device status.

For shared Ethernet adapter (**-net** flag), the following fields are supported:

- svea** Virtual Ethernet adapter.
- physloc** The physical location code of the server virtual adapter.
- sea** Shared Ethernet adapter.
- backing** Backing device.
- bdphysloc** The physical location code of the backing device.
- status** Shared Ethernet adapter status.

For NPIV (**-npiv** flag), the following fields are supported:

**name** Virtual fibre channel adapter name.

**physloc** The physical location code of the server virtual fibre channel adapter.

**clntid** Client logical partition ID.

**clntname** Client logical partition name.

**clntos** The operating system on the client logical partition.

**status** Virtual fibre channel adapter status.

**fc** Physical fibre channel adapter name.

**fcphysloc** The physical location of the fibre channel adapter.

**ports** Physical fibre channel port number.

**flags** Virtual fibre channel adapter flags.

**vfclient** Client virtual fibre channel adapter name.

**vfclientdrc** Client virtual fibre channel adapter used for Dynamic Reconfiguration Connection (DRC).

For active memory sharing (**-ams** flag), the following fields are supported:

**paging** Paging virtual target device.

**streamid** Stream ID.

**clientid** Client logical partition ID.

**status** Paging virtual target device status.

**redundancy** Redundancy usage for the paging VIOS logical partitions.

**backing** Backing device.

**poolid** Memory pool ID.

**vasi** Virtual Asynchronous Services Interface (VASI) device name.

**pager** Pager device name.

**vbsd** Virtual Block Storage Device (VBSD) name. Specifies a delimiter character to separate output fields.

**-fmt** *delimiter*

**-net**

Specifies the supplied device is a server virtual Ethernet adapter or if used with the **-all** flag all virtual Ethernet adapters and backing devices are displayed.

**-npiv**

Displays NPIV binding information.

**-plc** *PhysicalLocatoinCode*

Specifies the device physical location code of a server virtual adapter. This flag cannot be used with the **-vadapter** flag.

- type** Specifies the type of devices to display. The following fields are supported:
- disk** List physical backing devices.
  - lv** List logical volume backing devices.
  - optical** List optical backing devices.
  - tape** List tape backed devices.
  - file** List file backed devices.
  - file\_disk** List file backed disk devices.
  - file\_opt** List file backed optical devices.
  - net** List network devices. (This option cannot be used in combination with disk, lv, or optical)
- vadapter** Specifies the device name of a server virtual adapter. This flag cannot be used with the **-plc** flag.  
*VirtualServerAdapter*
- vtd** *PagingDevice*

## Output Field Definitions

| Field   | Description                     |
|---------|---------------------------------|
| SVSA    | Server Virtual SCSI Adaper      |
| Physloc | Physical Location Code          |
| VTD     | Virtual Target Device           |
| LUN     | Logical Unit Number             |
| SVEA    | Server Virtual Ethernet Adapter |
| SEA     | Shared Ethernet Adapter         |

## Exit Status

| Return code | Description                                                                            |
|-------------|----------------------------------------------------------------------------------------|
| 11          | No VTDs associated with device                                                         |
| 12          | No SEAs associated with device                                                         |
| 15          | Specified device is not a server virtual SCSI adapter                                  |
| 16          | Specified device is not a server virtual Ethernet adapter                              |
| 17          | Specified device in not in the AVAILABLE state                                         |
| 63          | Specified device is not a virtual fibre channel adapter on a server logical partition. |

## Examples

- To list all virtual target devices and backing devices mapped to the server virtual SCSI adapter **vnode2**, type:  

```
lsmap -vadapter vhost2
```

The system displays a message similar to the following:

```
SVSA Physloc Client Partition ID

vhost0 U9111.520.10004BA-V1-C2 0x00000004
```

```

VTD vtscsi0
Status Available
LUN 0x8100000000000000
Backing device vtd0-1
Physloc

VTD vtscsi1
LUN 0x8200000000000000
Backing device vtd0-2
Physloc

VTD vtscsi2
Status Available
LUN 0x8300000000000000
Backing device hdisk2
Physloc U787A.001.0397658-P1-T16-L5-L0

```

- To list the shared Ethernet adapter and backing device mapped to the virtual server Ethernet adapter **ent4**, type:

```
lsmap -vadapter
ent4 -net
```

The system displays a message similar to the following:

```

SVEA
Physloc

ent4 P2-I1/E1

SEA ent5
Backing device ent1
Status Available
Physloc P2-I4/E1

```

- To list the shared Ethernet adapter and backing device mapped to the virtual server Ethernet adapter **ent5** in script format separated by a : (colon), type:

```
lsmap -vadapter ent5 -fmt ":"
```

The system displays a message similar to the following:

```
ent5:ent8:ent2
```

- To list all virtual target devices and backing devices, where the backing devices are of type disk or lv, type:

```
lsmap -all -type disk lv
```

The system displays a message similar to the following:

```

SVSA Physloc Client Partition ID

vhost0 U9117.570.10D1B0E-V4-C3 0x00000000

VTD vtscsi0
Status Available
LUN 0x8100000000000000
Backing device hdisk0
Physloc U7879.001.DQD0KN7-P1-T12-L3-L0

VTD vtscsi2
Status Available
LUN 0x8200000000000000
Backing device lv04
Physloc

SVSA Physloc Client Partition
ID

```

```

vhost1 U9117.570.10D1B0E-V4-C4 0x00000000

VTD vtscsil
Status Available
LUN 0x8100000000000000
Backing device lv03
Physloc

```

5. To list NPIV mapping information, type:

```
lsmap -all -npiv
```

The system displays a message similar to the following:

```

Name Physloc CIntID CIntName CIntOS
=====
vfchost0 U8203.E4A.HV40026-V1-C12 1 HV-40026 AIX

Status:NOT_LOGGED_IN
FC name:fcs0 FC loc code:U789C.001.0607088-P1-C5-T1
Ports logged in:3
Flags:1 <not_mapped, not_connected>
VFC client name: VFC client DRC:

```

## Related Information

The `cfgdev` command, the `chdev` command, the `chpath` command, the `lsdev` command, the `lspath` command, the `mkpath` command, the `mkvdev` command, the `rmdev` command, the `rmpath` command.

## IVM Ismemdev command

### Purpose

Lists block storage devices on the Virtual I/O Server that are capable of being added to a shared memory pool for use as paging space devices.

### Syntax

```
lsmemdev -r avail [-m ManagedSystem] [-p <VIOS LPAR name> | --id <VIOS LPAR ID>] [--min
<minimum size in MB>] [--max <maximum size in MB>] [--filter "FilterData"] [-F "AttributeNames"]
[--header] [--help]
```

### Description

The `lsmemdev` command lists block storage devices on the Virtual I/O Server that are capable of being added to a shared memory pool for use as paging space devices. Only devices in the available state are displayed.

### Flags

|                                |                                                                                                                                                                                                                                                                                                                                                                        |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-r</b> <i>ResourceType</i>  | The type of resources to list:                                                                                                                                                                                                                                                                                                                                         |
|                                | <code>avail</code> : available resources                                                                                                                                                                                                                                                                                                                               |
| <b>-m</b> <i>ManagedSystem</i> | The name of the managed system. This attribute is optional because only one system is managed. The name can either be the user-defined name for the managed system, or be in the form <code>tttt-mmm*sssssss</code> , where <code>tttt</code> is the machine type, <code>mmm</code> is the model, and <code>sssssss</code> is the serial number of the managed system. |

**--filter** *FilterData*

The filters to apply to the resources to be listed. Filters are used to select which resources of the specified resource type are to be listed. If no filters are used, all of the resources of the specified resource type are listed. For example, specific logical partitions can be listed by using a filter to specify the names or IDs of the logical partitions to list. Otherwise, if no filter is used, all the logical partitions in the managed system are listed.

The filter data consists of filter name/value pairs, which are in comma-separated value (CSV) format. The filter data must be enclosed in quotation marks.

The format of the filter data is as follows:

```
"filter-name=value,filter-name=value,..."
```

Note that certain filters accept a comma-separated list of values, as follows:

```
"filter-name=value,value,...",..."
```

When a list of values is specified, the filter name/value pair must be enclosed in quotation marks. Depending on the shell being used, nested quotation marks might need to be preceded by an escape character, which is usually a backslash (\) character.

Unless otherwise indicated, multiple values can be specified for each filter.

#### Valid filter names:

types | storage\_pools | redundant

**-F** *AttributeName*

A delimiter separated list of attribute names for the desired attribute values to be displayed for each resource. If no attribute names are specified, values for all of the attributes for the resource are displayed.

When this option is specified, only attribute values are displayed. No attribute names are displayed. The attribute values displayed are separated by the delimiter which was specified with this option.

This option is useful when only attribute values are desired to be displayed, or when the values of only selected attributes are desired to be displayed.

#### Attribute names for partitions:

##### **device\_name**

The Virtual I/O Server name of a block storage device.

##### **phys\_loc**

If the device has a physical location code, this attribute is displayed by default with the physical location code as the value. If the device does not have a physical location code, this attribute is not displayed by default and has a blank value.

##### **redundant\_capable**

This attribute is not displayed by default, but if requested has a value of 0.

**size** Size of the block storage device in megabytes.

##### **storage\_pool**

If the device comes from a storage pool, this attribute is displayed by default with the storage pool as the value. If the device does not come from a storage pool, this attribute is not displayed by default and has a blank value.

**type** Type of the underlying device. Valid values follow:

- logical
- phys
- storage\_pool

**--header** Displays a header record, which is a delimiter-separated list of attribute names for the attribute values that are displayed. This header record is the first record displayed. This option is only valid when used with the **-F** option.

## Exit Status

This command has a return code of 0 on success.

## Security

This command is accessible by all users.

## Examples

To list the storage devices that are capable of being added to a memory pool, type:

```
lsmemdev -r avail
```

## Related Information

The **lshwres** and **chhwres** commands.

---

## Isnetsvc command

### Purpose

Displays the status of a network service.

### Syntax

```
Isnetsvc NetworkService
```

### Description

The **Isnetsvc** command displays the status of a network service. Use the *NetworkService* parameter to specify which service should have its status displayed.

### Parameters

|                       |                                                                                                                                                                                                                                                                                                                                                                    |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>NetworkService</i> | Specify on the following values:                                                                                                                                                                                                                                                                                                                                   |
| <b>cimserver</b>      | Returns the status of the <b>cimserver</b> daemon.                                                                                                                                                                                                                                                                                                                 |
| <b>inetd</b>          | Returns the status of the <b>inetd</b> subsystem. The <b>inetd</b> subsystem must be in the active state for the <b>telnet</b> and <b>ftp</b> daemons to be active. If the <b>inetd</b> subsystem is in the inoperative state, when you start the <b>startnetsvc</b> command with any of the supported network services it reactivates the <b>inetd</b> subsystem. |
| <b>ssh</b>            | Returns the status of the <b>ssh</b> daemon.                                                                                                                                                                                                                                                                                                                       |
| <b>telnet</b>         | Returns the status of the <b>telnet</b> daemon.                                                                                                                                                                                                                                                                                                                    |
| <b>ftp</b>            | Returns the status of the <b>ftp</b> daemon.                                                                                                                                                                                                                                                                                                                       |
| <b>xntpd</b>          | Returns the status of the <b>xntpd</b> daemon.                                                                                                                                                                                                                                                                                                                     |



## Exit Status

- 9 Invalid network service

## Examples

1. To list the status of the **inetd** subsystem, type:

```
lsnetsvc inetd
```

This command will return either active or not active.

2. To list the status of the **telnet** daemon, type:

```
lsnetsvc telnet
```

This command will return either active or not active.

3. To list the status of the **ftp** daemon, enter:

```
lsnetsvc ftp
```

This command will return either active or not active.

## Related Information

The entstat command, the hostmap command, the hostname command, the mktcpip command, the netstat command, the optimizenet command, the startnetsvc command, and the stopnetsvc command.

---

## Isnports command

### Purpose

Lists available ports that are capable of N\_Port ID Virtualization (NPIV) and related information.

### Syntax

```
Isnports [-fmt Delimiter] [-field Fieldname]
```

### Description

The Isnports command displays information for all the ports capable of NPIV. If you use the name of the adapter driver (for a particular physical port) in the **vfcmap** command, the **map\_port** attribute is set for a virtual fibre channel adapter on the server logical partition. This attribute maps the virtual fibre channel adapter on the server logical partition to a physical fibre channel port.

Without a flag, information is displayed in a column. Specifying the **-fmt** flag formats the output with a specified delimiter. If no NPIV ports are in the Virtual I/O Server logical partition, the error code E\_NO\_NPIV\_PORTS(62) is displayed.

### Flags

**-fmt** *delimiter*                      Specifies a delimiter character to separate output fields.

- field** Specifies a list of fields to be displayed. The following fields are supported:
- name** Physical port name
  - physloc** Physical port location code
  - fabric** Fabric support
  - tports** Total number of NPIV ports
  - aports** Number of available NPIV ports
  - swwpns** Total number of target worldwide port names supported
  - awwpns** Number of target worldwide port names available

## Output Field Definitions

| Field   | Description                                           |
|---------|-------------------------------------------------------|
| name    | Physical port name                                    |
| physloc | Physical port location code                           |
| fabric  | Fabric support                                        |
| tports  | Total number of NPIV ports                            |
| aports  | Number of available NPIV ports                        |
| swwpns  | Total number of target worldwide port names supported |
| awwpns  | Number of target worldwide port names available       |

## Exit Status

| Return code | Description                                 |
|-------------|---------------------------------------------|
| 62          | System does not have ports capable of NPIV. |

## Examples

- To list all the NPIV-capable ports, type:

```
lsnports
```

The system displays a message similar to the following:

```
Name physloc fabric tports aports swwpns awwpns
fcs0 U789D.001.DQDMLWV-P1-C1-T1 1 64 64 2048 2047
fcs1 U787A.001.DPM0WVZ-P1-C1-T2 1 63 62 504 496
```

- To list all the NPIV-capable ports in script format separated by two colons (::), type:

```
lsnports -fmt "::"
```

The system displays a message similar to the following:

```
fcs0::U789D.001.DQDMLWV-P1-C1-T1::1::64::64::2048::2047
fcs1::U787A.001.DPM0WVZ-P1-C1-T2::1::63::62::504::496
```

- To list specific fields (name, swwpns, and awwpns) of all the NPIV-capable ports, type:

```
lsnports -field name swwpns awwpns
```

The system displays a message similar to the following:

|      |        |        |
|------|--------|--------|
| name | swwpns | awwpns |
| fcs0 | 2048   | 2047   |

## Related Information

The `lsmap` and `vfcmmap` commands.

---

## Ispath command

### Purpose

Displays information about paths to a MultiPath I/O (MPIO) capable device.

### Syntax

```
lspath [-dev DeviceName] [-pdev Parent] [-status Status] [-conn Connection] [-field FieldName] [-fmt Delimiter]
```

```
lspath -dev DeviceName -pdev Parent [-conn Connection] -lsattr [-attr Attribute...]
```

```
lspath -dev DeviceName -pdev Parent [-conn Connection] -range -attr Attribute
```

### Description

The `lspath` command displays one of three types of information about paths to an MPIO capable device. It either displays the operational status for one or more paths to a single device, or it displays one or more attributes for a single path to a single MPIO capable device. The first syntax shown above displays the operational status for one or more paths to a particular MPIO capable device. The second syntax displays one or more attributes for a single path to a particular MPIO capable device. Finally, the third syntax displays the possible range of values for an attribute for a single path to a particular MPIO capable device.

### Displaying Path Status with the Ispath Command

When displaying path status, the set of paths to display is obtained by searching the device configuration database for paths that match the following criteria:

- The target device name matches the device specified with the **-dev** flag. If the **-dev** flag is not present, then the target device is not used in the criteria.
- The parent device name matches the device specified with the **-pdev** flag. If the **-pdev** flag is not present, then parent is not used in the criteria.
- The connection matches the connection specified with the **-conn** flag. If the **-conn** flag is not present, then connection is not used in the criteria.
- The path status matches status specified with the **-status** flag. If the **-status** flag is not present, the path status is not used in the criteria.

If none of the **-dev**, **-pdev**, **-conn**, or **-status** flags are specified, then all paths known to the system are displayed.

By default, this command will display the information in columnar form. When no flags are specified that qualify the paths to display, the format of the output is:

```
status device parent
```

Possible values that can appear for the status column are:

**enabled**

Indicates that the path is configured and operational. It will be considered when paths are selected for IO.

**disabled**

Indicates that the path is configured, but not currently operational. It has been manually disabled and will not be considered when paths are selected for IO.

**failed** Indicates that the path is configured, but it has had IO failures that have rendered it unusable. It will not be considered when paths are selected for IO.

**defined**

Indicates that the path has not been configured into the device driver.

**missing**

Indicates that the path was defined in a previous boot, but it was not detected in the most recent boot of the system.

**detected**

Indicates that the path was detected in the most recent boot of the system, but for some reason it was not configured. A path should only have this status during boot and so this status should never appear as a result of the **lspath** command.

## Displaying Path Attributes with the lspath Command

When displaying attributes for a path, the path must be fully qualified. Multiple attributes for a path can be displayed, but attributes belonging to multiple paths cannot be displayed in a single invocation of the **lspath** command. Therefore, in addition to the **-lsattr**, **-dev**, and **-pdev** flags, the **-conn** flags are required to uniquely identify a single path. For example:

- if only one path between a device and a specific parent, the **-conn** flag is not required
- if there are multiple paths between a device and a specific parent, the **-conn** flag is required

Furthermore, the **-status** flag is not allowed.

By default, this command will display the information in columnar form.

```
attribute value description user_settable
```

## Flags

|                                 |                                                                                                                                                                                                                                                                                                                                                                         |
|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-attr</b> <i>Attribute</i>   | Identifies the specific attribute to list. The ' <i>Attribute</i> ' is the name of a path specific attribute. When this flag is provided, only the identified attribute is displayed. Multiple instances of this flag may be used to list multiple attributes. If this flag is not specified at all, all attributes associated with the identified path will be listed. |
| <b>-lsattr</b>                  | Displays the attribute names, current values, descriptions, and user-settable flag values for a specific path.                                                                                                                                                                                                                                                          |
| <b>-dev</b> <i>Name</i>         | Specifies the logical device name of the target device whose path information is to be displayed.                                                                                                                                                                                                                                                                       |
| <b>-field</b> <i>FieldNames</i> | Specifies the list of fields to display. The following fields are supported: <ul style="list-style-type: none"> <li><b>status</b> Status of the path</li> <li><b>name</b> Name of the device</li> <li><b>parent</b> Name of the parent device</li> <li><b>conn</b> Path connection.</li> </ul>                                                                          |
| <b>-fmt</b> <i>Delimiter</i>    | Specifies a delimiter character to separate output fields.                                                                                                                                                                                                                                                                                                              |
| <b>-pdev</b> <i>Parent</i>      | Indicates the logical device name of the parent device of the path(s) to be displayed.                                                                                                                                                                                                                                                                                  |

- range** Displays the legal values for an attribute name. The **-range** flag displays the list attribute values in a vertical column as follows:
- ```
Value1
Value2
.
.
ValueN
```
- The **-range** flag displays the range attribute values as x...n(+i) where x is the start of the range, n is the end of the range, and i is the increment.
- status Status** The **-status Status** flag indicates the status to use in qualifying the paths to be displayed. When displaying path information, the allowable values for this flag are:
- enabled** Display paths that are **enabled** for MPIIO path selection.
 - disabled** Display paths that are **disabled** from MPIIO path selection.
 - failed** Display paths that are **failed** due to IO errors.
 - available** Display paths whose **path_status** is **PATH_AVAILABLE** (that is, paths that are configured in the system, includes **enabled**, **disabled**, and **failed** paths).
 - defined** Display paths whose **path_status** is **PATH_DEFINED**.
 - missing** Display paths whose **path_status** is **PATH_MISSING**.
- conn Connection** Indicates the connection information to use in qualifying the paths to be displayed.

Exit Status

Return code	Description
1	Invalid status value.

Examples

- To display, without column headers, the set of paths whose operational status is disabled, enter:

```
lspath -status disabled
```

The system will display a message similar to the following:

```
disabled hdisk1 scsi1
disabled hdisk2 scsi1
disabled hdisk23 scsi8
disabled hdisk25 scsi8
```

- To display the set of paths whose operational status is failed, enter:

```
lspath -status failed
```

The system will display a message similar to the following:

```
failed hdisk1 scsi1
failed hdisk2 scsi1
failed hdisk23 scsi8
failed hdisk25 scsi8
```

- If the target device is a SCSI disk, to display all attributes for the path to parent scsi0 at connection 5,0, use the command:

```
lspath -dev hdisk10 -pdev scsi0 -conn "5,0" -lsattr
```

The system will display a message similar to the following:

```
weight      1      Order of path failover selection true
```

Related Information

The `lsmmap` command, the `mkpath` command, the `chpath` command, and the `rmpath` command.

lspv command

Purpose

Displays information about a physical volume within a volume group.

Syntax

```
lspv [ -avail | -free | -size ][ -field Fieldname... ] [ -fmt Delimiter ]
```

```
lspv [ -map | -lv | -pv | -size] PhysicalVolume
```

```
lspv [ -map | -lv | -pv] PhysicalVolume [-field Fieldname ] [ -fmt Delimiter ]
```

Description

The `lspv` command displays information about the physical volume if the specific physical volume name is given. If the `lspv` command is run without any arguments, the default is to print every known physical volume in the system along with its physical disk name, physical volume identifiers (PVIDs), to which volume group, if any, the physical volume belongs, and the state if the volume group is active.

When the *PhysicalVolume* parameter is used, the following characteristics of the specified physical volume are displayed:

Physical volume	Name of the physical volume
Volume group	Name of volume group. Volume group names must be unique systemwide names and can be from 1 to 15 characters long.
PV Identifier	The physical volume identifier for this physical disk.
VG Identifier	The volume group identifier of which this physical disk is a member.
PVstate	State of the physical volume. If the volume group that contains the physical volume is activated with the activatevg command, the state is active, missing, or removed. If the physical volume is deactivated with the deactivatevg command, the state is varied off.
Allocatable	Allocation permission for this physical volume.
Logical volumes	Number of logical volumes using the physical volume.
Stale PPs	Number of physical partitions on the physical volume that are not current.
VG descriptors	Number of volume group descriptors on the physical volume.
PP size	Size of physical partitions on the volume.
Total PPs	Total number of physical partitions on the physical volume.
Free PPs	Number of free physical partitions on the physical volume.
Used PPs	Number of used physical partitions on the physical volume.
Free distribution	Number of free partitions available in each intra-physical volume section.
Used distribution	Number of used partitions in each intra-physical volume section.

Flags

-avail Lists only physical volumes that are available for use as a backing device for virtual SCSI. If the physical volume is assigned to a shared memory pool (to be used as a paging space device by a shared memory partition), it is not available and is not listed.

-free Lists only physical volumes that are available for use as a backing device. If the physical volume is already used as a backing device or is assigned to a shared memory pool (to be used as a paging space device by a shared memory partition), it is not available and is not listed.

-field *FieldNames* Specifies the list of fields to display. The following fields are supported if no physical volume is specified:

- pvname** Physical volume disk name
- pvid** Physical volume identifier
- vgname** Volume group the physical volume is in
- pvstate** Physical volume state (active, missing, removed, varied off)

The following fields are supported if a physical volume is specified:

- pvname** Physical volume disk name
- vgname** Volume group the physical volume is in
- pvid** Physical volume identifier
- vgid** Volume group identifier
- pvstate** Physical volume state (active, missing, removed, varied off)
- allocatable** Allocation permission for this physical volume.
- stale** Number of stale partitions on the disk
- ppsize** Physical partition size
- numlv** Number of logical volumes
- size** Number of physical partitions and total disk size
- vgds** Number of volume group descriptor areas within the volume group.
- free** Number of free partitions and free space
- pvused** Number of used partitions and used space
- maxreq** Maximum transfer size of physical volume
- freedist** Number of free partitions available in each intra-physical volume section.
- usedist** Number of used partitions in each intra-physical volume section
- hotspare**

-field *FieldNames*
(continued)

The following fields are supported if the **-pv** flag is specified:

range A range of consecutive physical partitions contained on a single region of the physical volume.

ppstate

The current state of the physical partitions: free, used, stale, or vgda

region The intra-physical volume region in which the partitions are located.

lvname

The name of the logical volume to which the physical partitions are allocated.

type

The type of the logical volume to which the partitions are allocated.

mount File system mount point for the logical volume, if applicable.

The following fields are supported if the **-map** flag is specified:

physical

Physical volume name and physical partition number.

logical Logical volume name and logical partition number. If mirrored the mirror number is also shown. If the partition is stale this is also shown.

The following fields are supported if the **-free** or **-avail** flag is specified:

pvname

Physical volume disk name.

pvid

Physical volume identifier

size

Size of the physical volume.

The following fields are supported if the **-size** flag and a physical volume are specified:

pvname

Physical volume disk name.

pvid

Physical volume identifier

size

Size of the physical volume.

Specifies a delimiter character to separate output fields.

-fmt *Delimiter*

-lv

Lists the following fields for each logical volume on the physical volume:

LVname

Name of the logical volume to which the physical partitions are allocated.

LPs

The number of logical partitions within the logical volume that are contained on this physical volume.

PPs

The number of physical partitions within the logical volume that are contained on this physical volume.

Distribution

The number of physical partitions, belonging to the logical volume, that are allocated within each of the following sections of the physical volume: outer edge, outer middle, center, inner middle and inner edge of the physical volume.

Mount Point

File system mount point for the logical volume, if applicable.

-map

Lists the following fields for each logical volume on the physical volume:
PVname:PPnum [LVname: LPnum [:Copynum] [PPstate]] Where:

PVname

Name of the physical volume as specified by the system.

PPnum Physical partition number.

LVname

Name of the logical volume to which the physical partitions are allocated. Logical volume names must be system-wide unique names, and can range from 1 to 64 characters.

LPnum Logical partition number. Logical partition numbers can range from 1 to 64,000.

Copynum

Mirror number.

PPstate Only the physical partitions on the physical volume that are not current are shown as stale.

-pv

Lists the following fields for each physical partition on the physical volume:

Range A range of consecutive physical partitions contained on a single region of the physical volume.

State The current state of the physical partitions: free, used, stale, or vgda.

Note: If a volume group is converted to a big vg format, it may be necessary to use some data partitions for volume group descriptor area. These partitions will be marked vgda.

Region The intra-physical volume region in which the partitions are located.

LVname

The name of the logical volume to which the physical partitions are allocated.

Type The type of the logical volume to which the partitions are allocated.

Mount Point

File system mount point for the logical volume, if applicable.

-size

Displays the size of one or all physical volumes in megabytes.

Exit Status

See "Exit status for Virtual I/O Server commands" on page 1.

Examples

1. To display the status and characteristics of physical volume **hdisk3**, type:

```
lspv hdisk3
```

2. To display all physical volumes in the system, type:

```
lspv
```

You should see output similar to the following:

```
hdisk0 0000000012345678 rootvg active
hdisk1 10000BC876543258 vg00 active
hdisk2 ABCD000054C23486 None
```

The previous example shows that physical volume **hdisk0** contains the volume group **rootvg**, and it is activated. Physical volume **hdisk1** contains the volume group **vg00**, and it is activated. Physical volume **hdisk2** does not contain an active volume group.

3. To display all physical volumes that can be virtual SCSI backing devices, type:

```
lspv -avail
```

Output similar to the following is displayed:

```
lspv -avail
NAME          PVID          SIZE(megabytes)
hdisk2        00c3e35c99c55ebd 7820
hdisk3        00c3e35c99c0a332 7820
hdisk4        00cbe8ddc00fbaad 7820
```

4. To display all physical volumes that can be virtual SCSI backing devices and are not currently a backing device, type:

```
lspv -free
```

Output similar to the following is displayed:

```
hdisk3 10000BC876543258 None None
hdisk4 ABCD000054C23486 None None
```

Related Information

The **migratepv** command.

IVM lsrefcode command

Purpose

Lists reference codes for partitions or the managed system. This command is valid only in an Integrated Virtualization Manager environment.

Syntax

To list reference codes for the managed system:

```
lsrefcode -r sys [ -n Number ] [ --filter "FilterData" ] [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

To list reference codes for partitions:

```
lsrefcode -r lpar [ -n Number ] [ --filter "FilterData" ] [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

Description

The **lsrefcode** command lists reference codes for partitions or the managed system.

Flags

- r** *ResourceType* The type of resources to list:
- **-r sys**: List reference codes for the managed system.
 - **Attributes**: refcode_num, time_stamp, refcode, word2, word3, word4, word5, word6, word7, word8, word9, fru_call_out_loc_codes
 - **Filters**: None
 - **-r lpar**: List reference codes for partitions.
 - **Attributes**: lpar_name, lpar_id, time_stamp, refcode, word2, word3, word4, word5, word6, word7, word8, word9, fru_call_out_loc_codes
 - **Filters**: { lpar_ids | lpar_names }
- n** *Number* The number of reference codes to list. The default is to list one. Reference codes are listed in order, with the most recent reference code first.

-m *ManagedSystem*

The name of the managed system. This attribute is optional because there is only one system to manage. The name might either be the user-defined name for the managed system, or be in the form `ttt-mmm*sssssss`, where `ttt` is the machine type, `mmm` is the model, and `sssssss` is the serial number of the managed system.

--filter *FilterData*

The filters to apply to the resources to be listed. Filters are used to select which resources of the specified resource type are to be listed. If no filters are used, then all of the resources of the specified resource type will be listed. For example, specific partitions can be listed by using a filter to specify the names or IDs of the partitions to list. Otherwise, if no filter is used, then all the partitions in the managed system will be listed.

The filter data consists of filter name/value pairs, which are in comma separated value (CSV) format. The filter data must be enclosed in double quotation marks.

The format of the filter data is as follows:

```
"filter-name=value,filter-name=value,..."
```

Certain filters accept a comma separated list of values, as follows:

```
"filter-name=value,value,...",..."
```

When a list of values is specified, the filter name/value pair must be enclosed in double quotation marks. Depending on the shell being used, nested double quotation marks might need to be preceded by an escape character, which is usually a `'\'` character.

Unless otherwise indicated, multiple values can be specified for each filter.

Valid filter names for -r lpar:

lpar_ids

ID of the partition to view

lpar_names

Name of the partitions to view

-F *AttributeName* A delimiter separated list of attribute names for the desired attribute values to be displayed for each resource. If no attribute names are specified, then values for all of the attributes for the resource will be displayed.

When this option is specified, only attribute values will be displayed. No attribute names will be displayed. The attribute values displayed will be separated by the delimiter which was specified with this option.

This option is useful when only attribute values are desired to be displayed, or when the values of only selected attributes are desired to be displayed.

Attribute Names:

fru_call_out_loc_codes
Location codes of the field replaceable unit related to the reference code

lpar_id Unique integer identifier for the partition

lpar_name
Name of the partition

refcode The ASCII reference code string

refcode_num
The sequence number of the reference code.

time_stamp
The time that the reference code was created in the format: MM/DD/YYYY HH:MM:SS where MM is the two digit month, DD is the two digit day, YYYY is the four digit year, HH is the two digit hour, MM is the two digit minute, and SS is the two digit second.

fru_call_out_loc_codes
The Field Replaceable Unit (FRU) numbers for the FRUs whose absence or failure caused this reference code to be received. This field might be used for other values.

--header Display a header record, which is a delimiter separated list of attribute names for the attribute values that will be displayed. This header record will be the first record displayed. This option is only valid when used with the -F option.

Exit Status

This command has a return code of 0 on success.

Security

This command is accessible by all users.

Examples

1. To list the current reference code for the managed system, type:
lsrefcode -r sys
2. To list the current reference code for all partitions, type:
lsrefcode -r lpar
3. To list the last 25 reference codes for partitions p1 and p2, only viewing the lpar_id and refcode attributes, type:
lsrefcode -r lpar -n 25 --filter "\lpar_names=p1,p2\" -F lpar_id,refcode

Isrep command

Purpose

Lists and displays information about the Virtual Media Repository.

Syntax

Isrep [-**field** *Fieldname*] [-**fmt** *Delimiter*]

Description

The **Isrep** command displays information about the Virtual Media Repository. The following information is displayed; the size of the repository and free space, parent storage pool, size and free space, and the name, size, associated virtual target device, and access state of all virtual optical media in the repository.

This command will provide full scripting support through the use of the **-field** and **-fmt** flags.

Flags

-field *FieldName*

size Total Size

free Free space

parent pool

Parent storage pool name

parent size

Parent storage pool size

parent free

Parent storage pool free space

name Number of backing device file

file size

Backing device file size

optical Virtual Target Device off the virtual optical device media is loaded in

access Media access, read-only (ro) or read-write (rw)

-fmt *Delimiter*

Specifies a delimiter character to separate output fields.

Examples

1. To display information about the Virtual Media repository and all virtual media with the repository, type the following command:

```
lsrep
```

The system displays output similar to the following:

SIZE(mb)	FREE(mb)	PARENT SIZE	PARENT FREE		
2039	299	18352	16304		
Name		File Size	Optical		Access
clientCD		640	vtopt3		ro
installDVD1		1000	vtopt16		rw
installDVD2		100	None		rw

2. To display only the parent storage pool size, type the following command:

```
lsrep -field "parent size"
```

```
Parent Size  
30624
```

lssp command

Purpose

Lists and displays information about storage pools.

Syntax

List all available storage pools

```
lssp [ -type PoolType [-field Fieldname ] [ -fmt Delimiter ]
```

Display information about a specific storage pool

```
lssp -detail | -bd [-sp StoragePool] [-field Fieldname ] [ -fmt Delimiter ]
```

Display the default storage pool

```
lssp -default
```

Description

The **lssp** command displays information about storage pools in the Virtual I/O Server. If no flags are specified, a list of all defined storage pools, their total size, free space, minimum allocation size, and number of backing devices contained in the pool and the type of pool is displayed. If the **-type** flag is specified only storage pools of the indicated type are displayed. If the **-detail** flag is specified, detailed information about the storage pool is displayed. If the **-bd** flag is specified, a list of all the backing devices in the specified (or default) storage pool is displayed along with their size and associated virtual target device, or None, and virtual host adapter, or None. If the **-default** flag is specified, the default storage pool is displayed.

This command will provide full scripting support through the use of the **-field** and **-fmt** flags.

Flags

-default Displays the default storage pool.

-field *FieldName*

The following fields are supported if no flags are specified:

pool Storage pool name
size Total size
free Free space
alloc Minimum allocation size
bds Number of backing devices
type Type of pool

The following fields are supported if the **-detail** flag is specified for a logical volume pool:

pvname Name of the physical volume.
pvid Physical volume identifier
size The size of the physical volume.

The following fields are supported if the **-detail** flag is specified for a file pool:

name Name of the parent storage pool.

The following fields are supported if the **-bd** flag is specified:

bdname Backing device name.
size The size of the logical volume.
vtd Virtual target device.
svsa Server virtual SCSI adapter.

-fmt *Delimiter*

-bd

-detail

-sp *StoragePool*

-type *PoolType*

Specifies a delimiter character to separate output fields.

Displays information about the backing devices in the storage pool.

Displays detailed information about the storage pool.

Specifies which storage pool to display information about.

Specifies the type of pool to list. The following fields are supported:

lvpool List only logical volume pools.

fbpool List only file pools.

Examples

1. To list all storage pools, type:

```
lssp
```

The system displays output similar to the following:

Pool	Size(mb)	Free(mb)	Alloc	Size(mb)	Bds	Type
PRODClient	30624	28576		32	0	LVPPOOL
rootvg	30656	14208		64	0	LVPPOOL
DEVClient	18352	18352		16	0	LVPPOOL
PRODClient_FBP	1016	985		32	3	FBPOOL
PRODClient_FBP2	1016	1005		32	1	FBPOOL

2. To display the default storage pool, type:

```
lssp -default
```

3. To display detailed information about the logical volume storage pool sp_sp00, type:

```
lssp -detail -sp sp_sp00
```

The system will display output similar to the following:

NAME	PVID	SIZE(megabytes)
hdisk3	00cdfd8c85bd4b2e	34624
hdisk2	00cdfd8c525d94a2	34624

- To display information about the backing devices in the logical volume storage pool rootvg, type:

```
lssp -bd -sp rootvg
```

The system displays output similar to the following:

NAME	SIZE(megabytes)	VTD	SVSA
lv01	96	vtscsi1	vhost0
lv02	64	vtscsi2	vhost0

Issvc command

Purpose

Lists the available agents.

Syntax

Issvc *AgentName*

Description

The **Issvc** command lists all the available agents that can be managed by the Virtual I/O Server command-line interface. If an *AgentName* is passed to the **Issvc** command, then a list of attributes with their configured values is displayed. These agents are managed by the **cfgsvc**, **startsvc**, and **stopsvc** commands.

Agent names

The following agents can be managed by the Virtual I/O Server.

DIRECTOR_agent	Lists the attributes and values for the IBM Systems Director agent.
ITM_base	Lists the attributes and values for the IBM Tivoli Monitoring agents.
ITM_premium	
ITM_cec	
TSM_base	Lists the attributes and values for the IBM Tivoli Storage Manager agent.
ITUAM_base	Lists the attributes and values for the IBM Tivoli Usage and Accounting Manager agent.
TPC	Lists the attributes and values for the TotalStorage Productivity Center agent.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

- To list available agents, type the following command:

```
lssvc
```
- To list values of attributes by agent name, type one or more of the following commands:

```
lssvc ITM_base
```

```
lssvc ITUAM_base
```

```
lssvc TSM_base
```

These commands produces output similar to the following:


```

$!ssvc ITM_base
HOSTNAME:tems_server
MANAGING_SYSTEM:hmc_console
RESTART_ON_REBOOT:TRUE

$!ssvc ITUAM_base
ACCT_DATA0:
ACCT_DATA1:
ISYSTEM:
IPROCESS:

$!ssvc TSM_base
SERVERNAME:
SERVERIP:
NODENAME:

```

- To list the mandatory attributes (for IBM TotalStorage Productivity Center) with their configured values, type the following command:

```
$!ssvc TPC
```

This command produces output similar to the following:

```

$!ssvc TPC
A:
S:
devAuth:
caPass:

```

Related Information

The **cfgsvc** command, the **startsvc** command, and the **stopsvc** command.

For more information about the various agents, see the following information:

- Configuring the IBM Systems Director agent
- Configuring the IBM Tivoli agents and clients on the Virtual I/O Server
- IBM Systems Director software
- IBM Tivoli software and the Virtual I/O Server

IVM Issvcevents command

Purpose

List attributes of console or serviceable events. This command is valid only in an Integrated Virtualization Manager environment.

Syntax

To list console events:

```
Issvcevents -t console [ -d NumberDays | -i NumberMinutes ] [ --filter "FilterData" ] [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

To list serviceable events:

```
Issvcevents -t hardware [ -d NumberDays | -i NumberMinutes ] [ --filter "FilterData" ] [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

To list field replaceable units (frus) for a specific serviceable events

```
Issvcevents -t fru --filter "FilterData" [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

To list comments for a specific serviceable event

```
lssvcevents -t comment --filter "FilterData" [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

To list service objects associated with a specific serviceable event

```
lssvcevents -t service_object -filter "FilterData" [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

To list the status of dynamic LPAR events:

```
lssvcevents -t dlpar [ --filter "FilterData" ] [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

Description

The `lssvcevents` command lists attributes of console or serviceable events.

Flags

`-t` *EventType*

The type of event to list:

console

Console events - events created by the applications

- **Attributes:** time, userid, pid, name, category, severity, time_bin, text
- **Filters:** severities, categories, name

hardware

Serviceable events

- **Attributes:** problem_num, pmh_num, refcode, status, first_time, last_time, sys_name, reporting_name, sys_mtms, reporting_mtms, enclosure_mtms, failing_mtms, text, firmware_fix, created_time, analyzing_sfp, refcode_extension, firmware_pkg_name, firmware_pkg_status, reporting_sfp_name, reporting_sfp_mtms, failing_sfp_mtms, severity, lpar_id, lpar_name, lpar_hostname, lpar_os_type, notification_type, notification_status, duplicate_count, analyzing_sfp_mtms, analyzing_sfp_name, called_home_sys_mtms, sys_log_id, platform_log_id, subsystem_id, creator_id, ipl_state, symptom, failing_lpar_id, failing_lpar_name, failing_lpar_os_type, fru_part_nums, fru_phys_locs, first_time_bin, last_time_bin, created_time_bin
- **Filters:** status, problem_nums, refcodes, fru_part_nums, fru_phys_locs, reporting_mtms, failing_mtm

fru

Field replaceable units (frus) for a serviceable event.

- **Attributes:** part_num, class, description, phys_loc, prev_replaced, replaced_time, serial_num, replacement_grp, ccin, logic_ctl_mtms, power_ctl_mtms, replaced_time_bin
- **Filters:** problem_nums (required)

comment

Comments for a serviceable event.

- **Attributes:** time,commenter,text
- **Filters:** problem_nums (required), status

service_object

Service objects for a serviceable event. These are not accessible to users without the DEUser or SRUser roles.

- **Attributes:** key, sys_log_id, notification_type, platform_log_id, severity, creator_id, refcode, subsystem_id, lpar_os_type, failing_mtms, text, lpar_id, lpar_name, lpar_hostname, first_time, last_time, duplicate_count, eed_ptr, first_time_bin, last_time_bin
- **Filters:** problem_nums (required), status

dlpar

Status of dynamic LPAR events - events used to synchronize resources that can be dynamically configured while the partition is active. These can be used to determine why the runtime and pending values of a resource are not synchronized. Typically, the last two status records are stored per partition per resource type.

- **Attributes:** lpar_id, resource_type, sequence_num, status_code, time, internal_rc, drmgr_cmd, drmgr_rc, drmgr_stdout, drmgr_stderr
- **Filters:** lpar_ids, resource_types

-d *NumberDays*

The number of previous days for which to view events. This might not be used with the **-i** flag, and is only applicable to **-t console**, and **-t hardware**. If this flag and the **-i** flag are omitted, the default is 7 days.

-i *NumberMinutes*

The number of minutes for which to view events. This might not be used with the **-d** flag, and is only applicable to **-t console**, and **-t hardware**.

-m *ManagedSystem*

The name of the managed system. This attribute is optional because there is only one system to manage. The name might either be the user-defined name for the managed system, or be in the form tttt-mmm*sssssss, where tttt is the machine type, mmm is the model, and sssssss is the serial number of the managed system.

--filter *FilterData*

The filters to apply to the resources to be listed. Filters are used to select which resources of the specified resource type are to be listed. If no filters are used, then all of the resources of the specified resource type will be listed. For example, specific partitions can be listed by using a filter to specify the names or IDs of the partitions to list. Otherwise, if no filter is used, then all the partitions in the managed system will be listed.

The filter data consists of filter name/value pairs, which are in comma separated value (CSV) format. The filter data must be enclosed in double quotation marks.

The format of the filter data is as follows:

```
"filter-name=value,filter-name=value,..."
```

Note that certain filters accept a comma separated list of values, as follows:

```
"filter-name=value,value,...",..."
```

When a list of values is specified, the filter name/value pair must be enclosed in double quotation marks. Depending on the shell being used, nested double quotation marks might need to be preceded by an escape character, which is usually a backslash character.

Unless otherwise indicated, multiple values can be specified for each filter.

Valid filter names for -t console:

severities, categories, name

Valid filter names for -t hardware:

status, problem_nums, refcodes, fru_part_nums, fru_phys_locs, reporting_mtms, failing_mtms

Valid filter names for -t fru:

problem_nums (required)

Valid filter names for -t comments and -t service_objects:

problem_nums (required), status

Valid filter names for -t dlpar:

lpar_ids, resource_types

-F *AttributeNames*

A delimiter separated list of attribute names for the desired attribute values to be displayed for each resource. If no attribute names are specified, then values for all of the attributes for the resource will be displayed.

When this option is specified, only attribute values will be displayed. No attribute names will be displayed. The attribute values displayed will be separated by the delimiter which was specified with this option.

This option is useful when only attribute values are desired to be displayed, or when the values of only selected attributes are desired to be displayed.

Attribute names:

analyzing_sfp

Name or MTMS of the service focal point system analyzing the event.

analyzing_sfp_mtms

MTMS of the service focal point system analyzing the event.

analyzing_sfp_name

Name of the service focal point system analyzing the event.

called_home_sys_mtms

MTMS of the system to which the event was called home.

category

Category or client type of the console event. Valid values:

- GUI: Web interface
- CLI: Command line interface
- AP: Access process

ccin CCIN of the FRU

class Class or type of the FRU

commenter

Name of the person adding a comment to the event

created_time

Time the event was created.

creator_id

The ID of the entity that created the event. Valid values:

- C: Hardware Management Console
- E: Service processor
- H: POWER hypervisor
- W: Power
- L: Partition firmware
- S: Licensed Internal Code for IBM i

description

Description of the FRU

drmgr_cmd

The command used for a dynamic LPAR event. The `drmgr` command is run on the client partition's operating system to synchronize a particular resource.

drmgr_rc

The return code for the command used for a dynamic LPAR event. The `drmgr` command is run on the client partition's operating system to synchronize a particular resource.

drmgr_stdout

The standard output for the command used for a dynamic LPAR event. The `drmgr` command is run on the client partition's operating system to synchronize a particular resource.

drmgr_stderr

The standard error for the command used for a dynamic LPAR event. The `drmgr` command is run on the client partition's operating system to synchronize a particular resource.

duplicate_count

Number of duplicates for this event.

enclosure_mtms

Enclosure MTMS

failing_lpar_id

Unique ID for the failing partition

failing_lpar_name
Name of the failing partition

failing_lpar_os_type
Operating system type of the failing partition

failing_mtms
MTMS of the failing system

failing_sfp_mtms
MTMS of the failing systems service focal point

firmware_fix
Indicates whether a firmware fix is available for the event

firmware_pkg_name
Package name of a possible firmware fix

firmware_pkg_status
Package status of a possible firmware fix

first_time
First time this event was reported

fru_part_nums
Part numbers of the FRUs

fru_phys_locs
Unique physical location codes of the FRUs

internal_rc
The return code for a dynamic LPAR event. This will be non zero only if the dynamic LPAR command was unexpectedly not able to be sent to the client partition.

ipl_state
State of the system when this event occurred

key Unique ID for a service object for a particular serviceable event.

last_time
Last time this event was reported

logic_ctl_mtms
MTMS of the unit that logically controls the unit that the FRU is located in

lpar_hostname
Hostname of the partition that created this event

lpar_id ID of the partition that created this event

lpar_name
Name of the partition that created this event

lpar_os_type
Operating system type of the partition that created this event

name Name of the application that created the console event

notification_status
The status of the notification type

notification_type

The notification type for the event. Valid values:

- Yes: Call home
- No: Customer notify

part_num

Part number of the FRU

phys_loc

Unique physical location code of the FRU

pid

Process ID of the process generating the console event

platform_log_id

Unique ID of the platform log for this event

pmh_num

PMH or tracking number

power_ctl_mtms

MTMS of the unit that power controls the unit that the FRU is located in

prev_replaced

Indicates if the FRU has been previously replaced

problem_num

A unique ID for the event

refcode Reference code for the event

refcode_extension

Extended reference code for the event

replaced_time

Time the FRU was replaced

replacement_grp

Replacement priority and grouping for the FRU. Valid values:

- H: Multiple high priority FRUs should be acted on as a group
- M: Medium priority FRUs should be acted on, one at a time, in the order given.
- A: Medium priority group A FRUs should be acted on as a group.
- B: Medium priority group B FRUs should be acted on as a group.
- C: Medium priority group C FRUs should be acted on as a group.
- L: Low priority FRUs should be acted on only after all other priority call-outs failed to resolve the problem.

reporting_mtms

MTMS of the reporting unit

reporting_name

Name of the reporting unit

reporting_sfp_mtms

MTMS of the service focal point reporting the event

reporting_sfp_name

Name of the service focal point reporting the event

resource_type

The resource type for the dynamic LPAR event. These are all the resources that can be dynamically reconfigured. Valid values:

- mem
- proc
- proc_units
- uncap_weight
- memory
- io_entitled_mem
- mem_weight

sequence_num

The sequence number for this dynamic LPAR event. Each resource type might have more than one entry. The sequence number distinguishes the entries, and increments over time.

serial_num

Serial number of the FRU

severity

Severity of the event. Valid values:

- 10: Recovered error, general (10)
- 20: Predictive error, general (20)
- 21: Predictive error, degraded performance (21)
- 22: Predictive fault might be corrected after platform re-IPL (22)
- 23: Predictive error, fault might be corrected after IPL, degraded performance (23)
- 24: Predictive error, loss of redundancy (24)
- 40: Unrecovered error, general (40)
- 41: Unrecovered error, bypassed with degraded performance (41).
- 44: Unrecovered error, bypassed with loss of redundancy (44)
- 45: Unrecovered error, bypassed with loss of redundancy and performance (45)
- 48: Unrecovered error, bypassed with loss of function (48)
- 60: Error on diagnostic test, general (60)
- 61: Error on diagnostic test, resource might produce incorrect result (61)

status Status of the event. Valid values:

- Open: Event is in the open state
- Closed: Event has been closed

status_code

Status code for the event. Valid values for dynamic LPAR events:

- 0: Synchronization successful
- 1: Synchronization in progress
- 2: Resource will not synchronize because the partition is a workload group participant
- 3: Resource will not synchronize because the partition communication state is not active
- 4: Resource will not synchronize because the partition does not support dynamic LPAR of this resource type
- 5: Resource will not synchronize because the partition is not in the Running state
- 6: Resource will not synchronize because the partition is unable to remove any more memory dynamically
- 7: Resource will not synchronize because the synchronization command failed to run for an unknown reason
- 8: Resource is not synchronized because the RMC command failed. The system will retry. If the partition is in the Running state with an active network connection, check the return code and contact your support representative.
- 9: Resource is not synchronized because the drmgr command on the partition failed. The system will retry. Check the return code, and the command output.
- 10: Resource will not synchronize because the requested assigned value is less than the current minimum. Restart your partition in order to complete the synchronization.
- 11: Resource will not synchronize because the requested assigned value is greater than the current maximum. Restart your partition in order to complete the synchronization.
- 12: Resource will not synchronize because the pending and current processing modes do not match. Restart your partition in order to complete the synchronization.
- 13: Resource will not synchronize because IVM is unable to determine dynamic LPAR capabilities of the logical partition.
- 255: Resource synchronization has not yet been attempted. It might take a few seconds depending on your system utilization before synchronization is attempted.

subsystem_id

The subsystem causing the event. Valid values include:

- 10 - 1F: Processor subsystem including internal cache
- 20 - 2F: Memory subsystem including external cache
- 30 - 3F: I/O subsystem (hub, bridge, bus)
- 40 - 4F: I/O adapter, device and peripheral
- 50 - 5F: CEC hardware
- 60 - 6F: Power/Cooling subsystem
- 70 - 79: Other subsystem
- 7A - 7F: Surveillance error
- 80 - 8F: Platform firmware
- 90 - 9F: Software
- A0 - AF: External environment

symptom

Symptom of the event

sys_log_id

Unique ID of the system log for the event

sys_mtms

MTMS of the system

text

Text of the event

time

Time of the console or dynamic LPAR event

time_bin

Time in milliseconds since Jan 1, 1970

userid User ID of the user that ran the command that caused the console event

--header

Display a header record, which is a delimiter separated list of attribute names for the attribute values that will be displayed. This header record will be the first record displayed. This option is only valid when used with the **-F** option.

Exit Status

This command has a return code of 0 on success.

Security

This command is accessible by all users, except for certain types and attributes as noted, which require SR User or DE User roles.

Examples

1. To list the serviceable events that occurred today, type:

```
lssvcevents -t hardware -d 0
```
2. To list the console events that occurred within the past 3 days, type:

```
lssvcevents -t console -d 3
```
3. To list all of the open serviceable events for the system, type:

```
lssvcevents -t hardware --filter "status=open"
```
4. To list the associated FRUs for a specific serviceable event, type:

```
lssvcevents -t fru  
--filter problem_nums=6013EFFF-205E9F22-4CC931E5-F892358-A0F6C1D6
```

Related Information

The `chsvcevent` command, and the `mksvcevent` command.

Issw command

Purpose

Lists installed software products.

Syntax

```
Issw [ -hist ]
```

Description

The `Issw` command displays information about installed file sets or file set updates. If the `-hist` parameter is not specified, the name, most recent level, state, and description of all file sets is displayed. Part information (usr, root, and share) is consolidated into the same listing. For formatted file sets, it displays the most recent maintenance level. Any interim fixes on the system are also displayed.

If the `-hist` flag is specified, installation and update history information is displayed.

Output Values

The following sections define terms used in several of the output fields. Note that not all output values are defined here. Only the ones that require explanation are defined.

State Values

The `state` field in the `Issw` output gives the state of the fileset on your system. It can have the following values:

State	Description
APPLIED	The specified fileset is installed on the system. The APPLIED state means that the fileset can be removed with the <code>updateios</code> command and the previous level of the fileset restored.
APPLYING	An attempt was made to apply the specified fileset, but it did not complete successfully, and cleanup was not performed.
BROKEN	The specified fileset or fileset update is broken and should be reinstalled before being used.
COMMITTED	The specified fileset is installed on the system. The COMMITTED state means that a commitment has been made to this level of the software. A committed fileset update cannot be rejected, but a committed fileset base level and its updates (regardless of state) can be removed by the <code>updateios</code> command.
EFIX LOCKED	The specified fileset was installed successfully and locked.
OBSOLETE	The specified fileset was installed with an earlier version of the operating system but has been replaced by a repackaged (renamed) newer version. Some of the files that belonged to this fileset have been replaced by versions from the repackaged fileset.
COMMITTING	An attempt was made to commit the specified fileset, but it did not complete successfully, and cleanup was not performed.
REJECTING	An attempt was made to reject the specified fileset, but it did not complete successfully, and cleanup was not performed.

Action Values

The **action** field in the **lssw** output identifies the installation action that was taken for the fileset. The following values may be found in this field:

Action	Definition
APPLY	An attempt was made to apply the specified fileset.
CLEANUP	An attempt was made to perform cleanup for the specified fileset.
COMMIT	An attempt was made to commit the specified fileset.
REJECT	An attempt was made to reject the specified fileset.

Status Values

The **status** field in the **lssw** output identifies the resultant status in the history of installation actions. The following values may be found in this field:

Status	Description
BROKEN	The fileset was left in a broken state after the specified action.
CANCELED	The specified action was canceled before it completed.
COMPLETE	The commitment of the fileset has completed successfully.

Flags

-hist Displays the installation and update history information.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1

Examples

1. To list all installed software, type:
`lssw`
2. To list installation and update history, type:
`lssw -hist`

Related Information

The **updateios** command, the **ioslevel** command, the **remote_management** command, the **oem_setup_env** command, and the **oem_platform_level** command.

IVM lssyscfg command

Purpose

List attributes of partitions, partition profiles, or the managed system. This command is valid only in an Integrated Virtualization Manager environment.

Syntax

To list partition attributes:

```
lssyscfg -r lpar [ --filter "FilterData" ] [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

To list partition profile attributes:

```
lssyscfg -r prof [ --filter "FilterData" ] [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

To list system attributes:

```
lssyscfg -r sys [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

Description

The `lssyscfg` command lists attributes of partitions, partition profiles, or the managed system.

Flags

-r <i>ResourceType</i>	The type of resources to list: lpar: Logical partition resources prof: Logical partition profile resources sys: Managed system resources
-m <i>ManagedSystem</i>	The name of the managed system. This attribute is optional because there is only one system to manage. The name may either be the user-defined name for the managed system, or be in the form <code>ttt-mmm*sssssss</code> , where <code>ttt</code> is the machine type, <code>mmm</code> is the model, and <code>sssssss</code> is the serial number of the managed system.
--filter <i>FilterData</i>	The filters to apply to the resources to be listed. Filters are used to select which resources of the specified resource type are to be listed. If no filters are used, then all of the resources of the specified resource type will be listed. For example, specific partitions can be listed by using a filter to specify the names or IDs of the partitions to list. Otherwise, if no filter is used, then all the partitions in the managed system will be listed. The filter data consists of filter name/value pairs, which are in comma separated value (CSV) format. The filter data must be enclosed in double quotes. The format of the filter data is as follows: "filter-name=value,filter-name=value,..." Note that certain filters accept a comma separated list of values, as follows: "filter-name=value,value,...",..." When a list of values is specified, the filter name/value pair must be enclosed in double quotes. Depending on the shell being used, nested double quote characters may need to be preceded by an escape character, which is usually a <code>'\'</code> character.

Unless otherwise indicated, multiple values can be specified for each filter.

Valid filter names for partitions:

lpar_names | lpar_ids - name or ID of the partitions to view
work_groups - work groups to which the partitions belong

Valid filter names for partition profiles:

lpar_names | lpar_ids: name or ID of the partition profiles
profile_names: profile names for the partitions
Note: This option is not valid when listing managed systems.

-F AttributeNames

A delimiter separated list of attribute names for the desired attribute values to be displayed for each resource. If no attribute names are specified, then values for all of the attributes for the resource will be displayed.

When this option is specified, only attribute values will be displayed. No attribute names will be displayed. The attribute values displayed will be separated by the delimiter which was specified with this option.

This option is useful when only attribute values are desired to be displayed, or when the values of only selected attributes are desired to be displayed.

Attribute names for partitions:

allow_perf_collection

Permission for the partition to retrieve shared processor pool utilization information. Valid values are:

- 0: do not allow authority
- 1: allow authority

auto_start

Valid values are:

- 0 - do not automatically start with system power on
- 1 - automatically start with system power on

boot_mode

Partition power on mode. This attribute is only available for AIX and Linux logical partitions. Valid values are:

- norm: normal
- dd: diagnostic with default boot list
- ds: diagnostic with stored boot list
- of: Open Firmware OK prompt
- sms: System Management Services
- null: not applicable

curr_lpar_proc_compat_mode

Displays the current negotiated compatibility mode. This is the mode that the partition operating system has negotiated, and will be less than or equal to the desired mode at logical partition boot time.

Note: It is possible that the mode will be greater than the desired mode, if the partition is powered off. The current value only changes when the logical partition restarts.

curr_profile

The current profile is always equal to the partition name

default_profile

The default profile is always equal to the partition name

desired_lpar_proc_compat_mode

Displays the requested compatibility mode.

dlpar_mem_capable

Indicates if the partition supports dynamic LPAR of memory. Valid values:

- 0 - no
- 1 - yes

dlpar_proc_capable

Indicates if the partition supports dynamic LPAR of processing resources. Valid values:

- 0 - no
- 1 - yes

ipl_source

The IPL source for the IBM i logical partition. This attribute is optional. Valid values:

- a
- b
- c
- d

logical_serial_num

A globally unique string for this partition

lpar_env

The operating environment for this partition Valid values:

- aixlinux: a partition type that supports AIX or Linux
- os400: an IBM i partition
- vioserver: a Virtual I/O Server partition

lpar_id unique integer identifier for the partition

lpar_keylock

Partition keylock position. Valid values:

- norm: normal keylock
- manual: manual keylock

mem_synchronized

The current and pending memory values for this partition are synchronized.

name Name of the partition

os_version

The version of the operating system running that is in the logical partition.

power_ctrl_lpar_ids

A list of partitions which have power control over this partition. Valid values:

- none: No partitions

proc_synchronized

The current and pending processing values for this partition are synchronized.

resource_config

Valid values:

- 0 - resources are not available to power on with system
- 1 - resources are available to power on with system

rmc_ipaddr

The IP address of the client partition. This IP address is used by RMC to connect to the client partition for dynamic LPAR.

rmc_state

The state of the RMC connection between the management partition and the client partition. The RMC connection is used primarily for dynamic LPAR.

Valid values:

- inactive
- active
- unknown
- none - RMC not configured. This partition has never been registered with RMC.

rmc_osshutdown_capable

Indicates if the partition supports shutdown via the RMC connection. This allows the management partition to safely shutdown the client partition via `chsysstate -o osshutdown` Valid values follow:

- 0 - no
- 1 - yes

state The current runtime state of the partition. Valid values follow:

- Not Activated
- Starting
- Running
- Shutting Down
- Error
- Open Firmware
- Not Available

uptime Partition uptime in seconds.

work_group_id

Valid values follow:

- none: do not participate in the workload management group
- 1: participate in the workload management group

Attribute names for partition profiles:**all_resources**

Valid values:

- 0: This partition will not own all physical resources on the system

alt_console_slot

The location of the physical I/O slot that contains the alternate console device for the IBM i logical partition. The default value is none.

alt_restart_device_slot

The location of the virtual I/O slot that contains the alternate restart device for the IBM i logical partition. If the load source slot is a value other than none, then this attribute is optional. Valid values are:

- Slot number (for virtual I/O)
- none

auto_start

Valid values:

- 0 - do not automatically start with system power on
- 1 - automatically start with system power on

boot_mode

Partition power on mode. Valid values are:

- norm - normal
- dd - diagnostic with default boot list
- ds - diagnostic with stored boot list
- of - Open Firmware OK prompt
- sms - System Management Services

conn_monitoring

Valid values:

- 0 - connection monitoring is disabled
- 1 - connection monitoring is enabled

console_slot

The location of the virtual I/O slot that contains the console device for the IBM i logical partition. Valid values are:

- Slot number (for virtual I/O)
- none

desired_io_entitled_mem

The amount of I/O entitled memory for a shared memory partition. This is the portion of memory that is reserved for I/O mappings.

- auto (automatically manage)
- *Number of megabytes*

If the value is auto, the entitlement is calculated based on the virtual I/O configuration of the partition. If the virtual I/O configuration is changed, the entitlement is updated automatically. If auto is not used, no automatic adjustments are made. The default value is auto.

desired_mem

The assigned megabytes of memory for this partition.

desired_procs

The assigned number of processors for this partition. In shared processing mode, this refers to virtual processors.

desired_proc_units

The assigned number of processing units for this partition.

hsl_pool_id

Whether the logical partition participates in the high-speed link (HSL) pool. The default value is 0 (does not participate).

io_slots

Comma separated list of I/O slots for the partition. Each item in this list has the format:

drc_index/slot_io_pool_id/is_required

Valid values for *is_required*:

- 0 - no
- 1 - yes

lhea_logical_ports

Comma-separated list of logical Host Ethernet Adapter (LHEA) logical ports, with each logical port having the following format:

adapter-ID/port-group/physical-port-ID
/logical-port-ID/allowed-VLAN-IDs

lhea_capabilities

Comma-separated list of LHEA capabilities, with each capability having one of the following formats:

adapter-ID/capability

adapter-ID/5/ieq/nieq/qp/cq/mr

The values for *ieq* (interruptible event queues), *niesq* (non-interruptible event queues), *qp* (queue pairs), *cq* (completion queues), and *mr* (memory regions) specify the resource amount in addition to the base minimum. Valid values are:

- 0 - minimum
- 1 - low
- 2 - medium
- 3 - high
- 4 - dedicated
- 5 - custom

load_source_slot

The location of the virtual I/O slot that contains the load source for the IBM i logical partition. If the *alt_restart_device_slot* has a value other than none, then this attribute is optional. Valid values are:

- Slot number (for virtual I/O)
- none

lpar_id Unique integer identifier for the partition.

lpar_io_pool_ids

Valid values are:

- none - this partition is not part of an I/O pool.

lpar_name

Name of the partition.

lpar_proc_compat_mode

The currently requested compatibility mode of the processor for the logical partition.

max_mem

The maximum megabytes of memory for this partition.

max_procs

The maximum number of processors for this partition. In shared processing mode, this refers to virtual processors. Valid values are:

- keep_idle_procs - never share processors
- share_idle_procs - share processors only when the partition is inactive
- share_idle_procs_active - share processors only when partition is active
- share_idle_procs_always - always share processors
- cap - capped mode
- uncap - uncapped mode

max_proc_units

The maximum number of processing units for this partition.

max_virtual_slots

Maximum number of virtual I/O adapter slots.

mem_mode

The partition memory mode.

- ded - dedicated memory
- shared - shared memory

If the memory mode is shared, then the logical partition cannot be assigned any physical I/O slots, the proc_mode attribute on the logical partition must be shared, and a memory pool must exist.

Note: The Virtual I/O Server logical partition only supports the dedicated memory mode.

mem_weight

The shared memory weight of the logical partition with shared memory. This flag is used for determining priority of logical partitions in a memory pool for distribution of memory.

min_mem

The minimum megabytes of memory for this logical partition.

min_procs

The minimum number of processors for this logical partition. In shared processing mode, this refers to virtual processors.

min_proc_units

The minimum number of processing units for this logical partition.

name Name of the profile.

op_console_slot

The location of the virtual I/O slot that contains the directly-attached operations console device for the IBM i logical partition. The default value is none.

os_type

The operating system environment for this logical partition. Valid values:

- aixlinux: an RPA partition type which supports AIX or Linux
- os400: an IBM i logical partition
- vioserver: a Virtual I/O Server partition

paging_device

The paging space device to use if a memory pool is used. A paging device is a block storage device that has been added to the memory pool and is not designated as a paging device for any other logical partition. If the paging_device value is a blank string, no paging device is currently assigned.

power_ctrl_lpar_ids

A list of logical partitions which have power control over this logical partition. Valid values are:

- none - No partitions

power_ctrl_lpar_names

A list of logical partitions which have power control over this logical partition. Valid values are:

- none - No partitions

primary_paging_vios_id

The ID of the primary paging Virtual I/O Server (VIOS) partition that provides access to the paging space devices for the shared memory partitions. A paging VIOS partition is a VIOS logical partition that is assigned to the shared memory pool.

primary_paging_vios_name

The name of the primary paging VIOS partition that provides access to the paging space devices for the shared memory partitions. A paging VIOS partition is a VIOS logical partition that is assigned to the shared memory pool.

proc_mode

Valid values:

- ded: dedicated processor mode
- shared: shared processor mode

secondary_paging_vios_id

The ID of the secondary paging Virtual I/O Server (VIOS) partition that provides access to the paging space devices for the shared memory partitions. A paging VIOS partition is a VIOS logical partition that is assigned to the shared memory pool.

secondary_paging_vios_name

The name of the secondary paging VIOS partition that provides access to the paging space devices for the shared memory partitions. A paging VIOS partition is a VIOS logical partition that is assigned to the shared memory pool.

sharing_mode

Valid values:

- share_idle_procs
- proc_mode=ded
- share_idle_procs_active
- share_idle_procs_always
- proc_mode=shared
- cap
- uncap

uncap_weight

A weighted average of processing priority when in uncapped sharing mode. The smaller the value, the lower the weight. Valid values are: 0 - 255.

virtual_eth_adapters

Comma-separated list of virtual Ethernet adapters, with each adapter having the following format: slot_number/is_ieee/port_vlan_id/additional_vlan_ids/is_trunk/is_required

All 5 '/' characters must be present, but optional values may be omitted. Optional values are additional-vlan-IDs, and is-trunk. Valid values for is_ieee, is_trunk, and is_required:

- 0 - no
- 1 - yes

For example, 4/0/2//0/0 specifies a virtual Ethernet adapter with a virtual slot number of 4, is not IEEE 802.1Q enabled, has a port virtual LAN ID of 2, no additional virtual LAN IDs, it is not a trunk adapter, and is not required.

A value of none indicates that there are no virtual Ethernet adapters.

virtual_fc_adapters

Comma-separated list of virtual fibre channel adapters. Each item in the list has the following format:

```
virtual slot num/adapter_type/remote_lpar_id/  
remote_lpar_name/remote_slot_num/wwpn_list/is_required
```

virtual_opti_pool_id

The location of the virtual I/O slot that contains the directly-attached operations console device for the IBM i logical partition. The default value is 0 (does not participate).

virtual_scsi_adapters

Comma-separated list of virtual SCSI adapters. Each item in the list has the following format:

```
slot_num/adapter_type/remote_lpar_id/remote_lpar_name/  
remote_slot_num/is_required
```

virtual_serial_adapters

Comma-separated list of virtual serial adapters. Each item in the list has the following format:

```
slot_num/adapter_type/supports_hmc/remote_lpar_id/  
remote_lpar_name/remote_slot_num/is_required
```

The attribute names are not present in the list, just their values are present. If an attribute is optional and is not to be included, then no value would be specified for that attribute. For example, 0/server/1/any//any/1 specifies a virtual server serial device that has a virtual slot number of 0, supports HMC, supports any remote partition, supports any remote slot, and is required.

A value of none indicates that there are no virtual serial adapters.

Valid values for adapter_type:

- client: client adapter
- server: server adapter, valid for Virtual I/O Server partitions only

Valid values for supports_hmc:

- 0 - no
- 1 - yes

Valid values for is_required:

- 0 - no
- 1 - yes

Attribute names for the managed system:

active_lpar_share_idle_procs_capable

Indicates if the platform supports setting the share while active sharing_mode.

Valid values:

- 0 - not capable
- 1 - capable

active_mem_sharing_capable

The capability of the managed system to use a shared memory pool. Valid values follow:

- 0 - Not capable
- 1 - Capable

capabilities

Displays a comma-separated list of capabilities.

cod_mem_capable

Valid values:

- 0- not capable of memory Capacity on Demand
- 1 - capable of memory Capacity on Demand

cod_proc_capable

Valid values:

- 0- not capable of processor Capacity on Demand
- 1 - capable of processor Capacity on Demand

config_version

The version of the configuration data in platform firmware of the current logical partition .

curr_configured_max_lpars

The current maximum number of logical partitions supported by the management logical partition

This option is deprecated. Instead use:

```
lshwres -r virtualio --subtype slot --level lpar
```

with attribute "curr_max_virtual_slots" value for partition 1

dlpar_mem_capable

Valid values:

- 0: Changes do not take effect until the next reboot of the logical partition or platform.
- 1: Changes take effect immediately.

lhea_capable

Indicates if the managed system has one or more Host Ethernet Adapters.

lpar_avail_priority

The priority of the partition to maintain its entitled processors. If a processor failure occurs, processing resources will be removed first from the lowest priority partition.

Note: The Virtual I/O Server partition must have a higher priority than any other partition on the system.

lpar_avail_priority_capable

Specifies if the platform supports the lpar_avail_priority attribute. Valid values:

- 0 - not capable
- 1 - capable

lpar_comm_default

Indicates if the **lpar_comm_ipaddr** is using the default IP address, or if the user has manually set this using **chsyscfg**. Valid values:

- 0: User has manually set the IP address
- 1: Default IP address is used. This is the first IP address configured on your system as reported by `lstcpip -interfaces`

lpar_comm_ipaddr

The IP address through which client partitions will communicate with the management partition. This is used primarily for dynamic LPAR. It is defaulted to the first IP address configured on your system, but can be manually set if desired.

Note: This attribute supports multiple IP addresses using a comma-separated list.

lpar_proc_compat_mode_capable

The supported compatibility modes of the processor for the logical partition. Valid values:

- 0: The managed system does not support setting the processor compatibility mode for a partition.
- 1: The managed system supports setting the processor compatibility mode for a partition.

lpar_proc_compat_modes

A comma separated list of compatibility modes that the managed system supports.

max_lpars
Maximum number of partitions supported by firmware.

max_power_ctrl_lpars
Maximum number of power controlling partitions per controlled partition.

max_vscsi_remote_lpar_id
Indicates the largest partition ID of the remote partition with a virtual SCSI server adapter.

max_micro_lpar_id
Indicates largest partition ID of a partition using the MicroPartition technology.

micro_lpar_capable
Valid values:
0: not capable of creating shared processor partitions
1: capable of creating shared processor partitions

mfg_default_config
Specifies whether or not the system is in the manufacturing default partition configuration.

Valid values:
0: No
1: Yes

name Name for the managed system.

pend_configured_max_lpars
The maximum number of partitions supported by the management partition after the next restart.

This option is deprecated. Instead use:
lshwres -r virtualio --subtype slot --level lpar

with attribute "pend_max_virtual_slots" value for partition 1.

pend_lpar_config_state
Valid values:

enabled: The logical partition configuration data will be enabled during the next restart operation.
disabled: The logical partition configuration data will be disabled during the next restart operation.
cleared: The logical partition configuration data will be cleared to manufacturing defaults during the next restart operation.

power_off_policy
Valid values:
• 0- Power off the managed system after all partitions are powered off.

os400_capable
Indicates that the platform supports IBM i logical partitions. Valid values:
• 0: not capable of running IBM i
• 1: capable of running IBM i

serial_num
Serial number for the managed system.

service_lpar_id
ID of the partition with platform service authority.

- service_lpar_name**
Name of the partition with platform service authority.
- state** Valid values:
- Operating - The managed system is running.
- sys_time**
The UTC time of system firmware in the format *month/day/year hour:minute:second*.
- type_model**
Type and model for the managed system
- vet_activation_capable**
Whether the platform supports PowerVM™ Editions system technologies activation. Valid values:
- 0 - not capable
 - 1 - capable
- virtual_fc_capable**
A value indicating whether the managed system supports a virtual fibre channel. Valid values follow:
- 0: The managed system does not support a virtual fibre channel.
 - 1: The managed system supports a virtual fibre channel.

--header Display a header record, which is a delimiter separated list of attribute names for the attribute values that will be displayed. This header record will be the first record displayed. This option is only valid when used with the **-F** option.

Exit Status

This command has a return code of 0 on success.

Security

This command is accessible by all users.

Examples

1. To list the attributes for the managed system, type:
lssyscfg -r sys
2. To list only the user-defined name, machine type and model, and serial number for the managed system, type:
lssyscfg -r sys -F name,type_model,serial_num
3. To list all partitions and only display attribute values for each partition following a header of attribute names, type:
lssyscfg -r lpar -F --header
4. To list the partitions named lpar1, lpar2, and lpar3, type:
lssyscfg -r lpar --filter \"lpar_names=lpar1,lpar2,lpar3\"
5. To list the partition profile for partition lpar2, type:
lssyscfg -r prof --filter lpar_names=lpar2
6. To view the desired compatibility mode in the profile, type:
lssyscfg -r prof -F lpar_proc_compat_mode

Related Information

The `chsyscfg` command, the `mksyscfg` command, and the `rmsyscfg` command.

IVM `Issysconn` command

Purpose

List connection information for systems. This command is valid only in an Integrated Virtualization Manager environment.

Syntax

To list the service processor network connection information:

```
Issysconn -r all [ -F "AttributeNames" ] [ --header ] [ -m ManagedSystem ]
```

Description

The `Issysconn` command lists the service processor network connection information.

Flags

- | | |
|-------------------------------|--|
| <code>-r ResourceType</code> | The type of resources to list:

all - Lists all connections
Attributes: resource_type, type_model_serial_num, sp, side, ipaddr, alt_ipaddr, state, eth_loc_code, alt_eth_loc_code
Filters: None |
| <code>-m ManagedSystem</code> | The name of the managed system. This attribute is optional because there is only one system to manage. The name may either be the user-defined name for the managed system, or be in the form <code>ttt-mmm*sssssss</code> , where <code>ttt</code> is the machine type, <code>mmm</code> is the model, and <code>sssssss</code> is the serial number of the managed system. |

-F *AttributeNames*

A delimiter separated list of attribute names for the desired attribute values to be displayed for each resource. If no attribute names are specified, then values for all of the attributes for the resource will be displayed.

When this option is specified, only attribute values will be displayed. No attribute names will be displayed. The attribute values displayed will be separated by the delimiter which was specified with this option.

This option is useful when only attribute values are desired to be displayed, or when the values of only selected attributes are desired to be displayed.

The following attributes are available:

resource_type

Indicates the resource type. This attribute always reads 0.

type_model_serial_num

The type-model and serial number of the system in the form ttt-mmm*sssssss, where ttt is the machine type, mmm is the model, and ssssssss is the serial number of the managed system.

sp The type of service processor. This attribute always reads unavailable.

side The current side of the service processor. This attribute always reads unavailable.

ipaddr The IP Address of the first Ethernet device on the service processor.

alt_ipaddr

The IP Address of the second Ethernet device on the service processor.

state The connection state to the service processor. This attribute always reads No Connection.

eth_loc_code

The physical location code of the first Ethernet device on the service processor.

alt_eth_loc_code

The physical location code of the second Ethernet device on the service processor.

--header

Display a header record, which is a delimiter separated list of attribute names for the attribute values that will be displayed. This header record will be the first record displayed. This option is only valid when used with the -F option.

Exit Status

This command has a return code of 0 on success.

Security

This command is accessible by all users.

Examples

1. To list all system connections, type:
lssysconn -r all

Related Information

The lssyscfg command.

IVM Isssysplan command

Purpose

Lists the system plan files in the Integrated Virtualization Manager directory where system plan files are kept.

Syntax

```
Isssysplan [-f <file name> -t plan] [-F [<attribute names>] [--header]] [--help]
```

Description

The **Isssysplan** command lists the system plan files in the Integrated Virtualization Manager directory where system plan files are kept.

Flags

-f <i>-tplan</i>	Specifies the name of the system plan file and provides the following details: sys_name, type_model, and ivm_deployable
-F	A delimiter-separated list of attribute names for the attribute values to be displayed for each file. If no attribute names are specified then the values for all the attributes for each file are displayed. When this option is specified, only attribute values are displayed; no attribute names are displayed. The attribute values displayed are separated by the delimiter that was specified with this option. The attributes that can be listed are as follows: name, description, source, version, and date.
--header	Displays a header record, which is a delimiter-separated list of attribute names for the attributes that are displayed. The header record is the first record displayed. This option is only valid with the -f option.
--help	Displays the help text for this command and exit.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

1. To list the system plan files on the Integrated Virtualization Manager system, type the following command:

```
Isssysplan
```
2. To list the system plan files following a header, type the following command:

```
Isssysplan -F --header
```
3. To list the name and creation date of the system plan file, type the following command:

```
Isssysplan -F name,date
```
4. To list the names and descriptions of the system plan following a header, type the following command:

```
Isssysplan -F name,description --header
```

Related Information

The **deploysysplan** command, **mksysplan** command, and the **rmsysplan** command.

IVM Issysplanres command

Purpose

Lists the system plan resources that are defined on the Integrated Virtualization Manager (IVM).

Syntax

```
Issysplanres -r osinstall [-F "<attributes>" [--header]] [-help]
```

Description

The Issysplanres command lists the system plan resources that are defined on the IVM. These resources can be used when deploying system plans from this IVM.

Flags

-r	Specifies the type of system plan resources to list. To list the installation resources for the operating system, the only valid value is osinstall .
-F	Lists the names of the attributes in delimiter-separated list. If no attribute names are specified, all attributes are listed.
--header	Displays a header record, which is a delimiter-separated list of attribute names for the attributes that are displayed. The header record is the first record displayed. This option is only valid with the -F flag.
-help	Displays the help text for this command and exit.

Exit Status

The following exit values are returned:

0	Successful completion.
>0	Invalid flag, argument, or command failure.

Examples

1. To list all of the system plan resources that have an **osinstall** type on this Integrated Virtualization Manager, type:

```
Issysplanres -r osinstall
```
2. To use a header record and list only the names and descriptions of system plan resources that have an **osinstall** type defined on this IVM:

```
Issysplanres -r osinstall -F name,description --header
```

Related Information

The defsysplanres, deploysysplan, Issysplan, and rmsysplanrescommands.

Istcpip command

Purpose

Displays the Virtual I/O Server TCP/IP settings and parameters.

Syntax

`lstcpip -interfaces [-fmt delimiter]`

`lstcpip [-state] [-num] [-field] [-fmt delimiter]`

`lstcpip [-num] [-routtable] [-routinfo] [-state] [-arp]`

`lstcpip -stored`

`lstcpip -adapters`

`lstcpip [-sockets] [-family {inet | inet6 | unix}]`

`lstcpip -namesrv`

`lstcpip [-state [-field FieldName ...]] | [-routtable [-field FieldName ...]] [-fmt delimiter]`

`lstcpip -hostname`

Note: If IPv6 is configured on the Virtual I/O Server, the **lstcpip** command lists an IPv6 address. Due to its size, the IPv6 address spans both the Network and Address fields of the output screen.

Description

The **lstcpip** command displays the current and stored TCP/IP setting such as IP address, routing table, sockets, name server settings, and so forth.

Flags

-adapters	Lists Ethernet adapters on the system.
-arp	Displays the current ARP table entries.
-family	Specifies the INET, INET6, or UNIX [®] socket family.
-field	Specifies a list of fields to display.
-fmt	Divides output by a user-specified delimiter.
-hostname	Displays the system hostname.
-interfaces	Displays all of the interfaces configured on the system. Specifically, it displays their addresses, network masks, states, and mac addresses.
-namesrv	Lists DNS name servers in search order and domain name.
-num	Displays numeric output, rather than trying to resolve host names.
-routtable	Displays the routing tables.
-routinfo	Displays the routing tables, including the user-configured and current costs of each route.
-sockets	Displays information about currently open sockets.
-state	Displays the current state of all configured interfaces.
-stored	Displays stored TCP/IP configuration, which will be applied when the system starts. It will list interface IP addresses, any defined static routes, host names, and DNS info.

Examples

1. To list the Virtual I/O Server TCP/IP configuration, type:

```
lstcpip -stored
```

2. To list the current routing table, type:

```
lstcpip -routtable
```

3. To list open inet sockets, type:

```
lstcpip -sockets -family inet
```

4. To display the states of all interfaces using a delimiter, type:

```
lstcpip -states -fmt "/"
```

This command produces output similar to the following:

```
sit0/1480/link#2/9.3.126.60/0/0/0/0/0
sit0/1480/commo126060./austin.ibm.com/0/0/0/0/0
en2/1500/link#3/0.9.6b.6e.e3.72/871825/0/16305/1/0
en2/1500/9.3.126/commo126060.austi/871825/0/16305/1/0
en2/1500/fe80::209:6bff:fe6e:e372/871825/0/16305/1/0
lo0/16896/link#1/5013/0/5019/0/0
lo0/16896/127/localhost/5013/0/5019/0/0
lo0/16896/::1/5013/0/5019/0/0
```

5. To display all interface information, type:

```
lstcpip -interfaces
```

This command produces output similar to the following:

Name	Address	mask/Pfx	State	MAC
en2	fe80::209:6bff:fe6e:e372	64	up	00:09:6b:6e:e3:72
en3	-	-	down	00:09:6b:6e:e3:73
et2	-	-	down	00:09:6b:6e:e3:72
et3	-	-	down	00:09:6b:6e:e3:73
en4	-	-	down	0e:f0:c0:00:30:04

lsuser command

Purpose

Displays user account attributes.

Syntax

```
lsuser [ ALL | Name[, Name ] ...]
```

Description

The **lsuser** command displays the user account attributes. You can use this command to list all attributes of all the system user accounts or all the attributes of specific user accounts. If you specify more than one user account, each user account is separated by a comma. If you do not specify any user accounts, the attributes of all user accounts are displayed.

The **lsuser** command lists each user's attributes on one line. It displays attribute information as *Attribute=Value* definitions, each separated by a blank space.

Exit Status

See "Exit status for Virtual I/O Server commands" on page 1.

Security

This command can be run by any users. However, user attributes are only displayed for the **padmin** user.

Examples

1. To list all users on the system, type:
lsuser

The system displays output similar to the following for the padmin user:

```
padmin roles=PAdmin account_locked=false expires=0 histexpire=0
histsize=0 loginretries=0 maxage=0 maxexpired=-1 maxrepeats=8 minage=0
minalpha=0 mindiff=0 minlen=0 minother=0 pldwarntime=0
sally roles=DEUser account_locked=false expires=0 histexpire=0
histsize=0 loginretries=0 maxage=0 maxexpired=-1 maxrepeats=8 minage=0
minalpha=0 mindiff=0 minlen=0 minother=0 pldwarntime=330
henry roles=DEUser account_locked=false expires=0 histexpire=0
histsize=0 loginretries=0 maxage=0 maxexpired=-1 maxrepeats=8 minage=0
minalpha=0 mindiff=0 minlen=0 minother=0 pldwarntime=330
admin1 roles=Admin account_locked=false expires=0 histexpire=0
histsize=0 loginretries=0 maxage=0 maxexpired=-1 maxrepeats=8 minage=0
minalpha=0 mindiff=0 minlen=0 minother=0 pldwarntime=330
deuser1 roles=DEUser account_locked=false expires=0 histexpire=0
histsize=0 loginretries=0 maxage=0 maxexpired=-1 maxrepeats=8 minage=0
minalpha=0 mindiff=0 minlen=0 minother=0 pldwarntime=330
sadan roles=Admin account_locked=false expires=0 histexpire=0
histsize=0 loginretries=0 maxage=0 maxexpired=-1 maxrepeats=8 minage=0
minalpha=0 mindiff=0 minlen=0 minother=0 pldwarntime=330 registry=LDAP
SYSTEM=LDAP sruser1 roles=SRUser,RunDiagnostics
account_locked=false expires=0 histexpire=0
histsize=0 loginretries=0 maxage=0 maxexpired=-1 maxrepeats=8 minage=0
minalpha=0 mindiff=0 minlen=0 minother=0 pldwarntime=330
view1 roles=ViewOnly account_locked=false expires=0 histexpire=0
histsize=0 loginretries=0 maxage=0 maxexpired=-1 maxrepeats=8 minage=0
minalpha=0 mindiff=0 minlen=0 minother=0 pldwarntime=330
```

The system displays the following information for other users:

```
padmin roles=PAdmin
sally roles=DEUser
henry roles=DEUser
admin1 roles=Admin
deuser1 roles=DEUser
sadan roles=Admin
sruser1 roles=SRUser
view1 roles=ViewOnly
```

2. To display the attributes of user admin1, type the following command:
`lsuser admin1`
3. To display the attributes of user admin1 and user admin2, type the following command:
`lsuser admin1, admin2`

Related Information

The `chuser` command, the `mkuser` command, the `rmuser` command, and the `passwd` command.

IVM `lsvet` command

Purpose

List Capacity on Demand advanced functions activation information.

Syntax

```
lsvet -t {code | hist} -m managed-system [-F [attribute-names] [--header]] [--help ]
```

Description

The `lsvet` command lists activation information for Capacity on Demand advanced functions. Capacity on Demand advanced functions include PowerVM Editions and Enterprise Enablement. Capacity on Demand advanced functions are sometimes referred to as Virtualization Engine systems technologies.

Flags

-t	The IP address or host name of the managed system. Use code for the Virtualization Engine systems technologies activation codes, and hist for the Virtualization Engine systems technologies activation history log. Attribute name time_stamp=08/30/2007 00:16:28,entry=[VIOSI0500040A-0336] IBM i processor capacity limit enabled.
-m	The name of the managed system for which information is to be listed. The name may either be the user-defined name for the managed system, or be in the form <i>ttt-mmm*sssssss</i> , where <i>ttt</i> is the machine type, <i>mmm</i> is the model, and <i>sssssss</i> is the serial number of the managed system. The <i>ttt-mmm*sssssss</i> form must be used if there are multiple managed systems with the same user-defined name.
-F	A delimiter separated list of attribute names representing the desired attribute values to display. If this option is specified without any attribute names, then all of the attributes will be displayed. When this option is specified, only attribute values will be displayed. No attribute names will be displayed. The attribute values displayed will be separated by the delimiter which was specified with this option. This option is useful when only attribute values are desired to be displayed, or when the values of only selected attributes are desired to be displayed.
--header	Display a header record, which is a delimiter separated list of attribute names for the attribute values that will be displayed. This header record will be the first record displayed. This option is only valid when used with the -F option.
--help	Display the help text for this command and exit.

Exit Status

This command has a return code of 0 on success.

Examples

1. To display the activation code generation information, type the following command:

```
lsvet -m sys1 -t code
```

2. To display the activation history log, type the following command:

```
lsvet -m 9117-570*1001213 -t hist
time_stamp=02/07/2006 19:52:03,entry=HSCL0421 POWER Hypervisor code entered.
time_stamp=02/07/2006 19:52:03,entry=HSCL0403 Virtual I/O server
                                     capability enabled.
time_stamp=02/07/2006 19:52:03,entry=HSCL0405 Micro-partitioning
                                     capability enabled.
time_stamp=02/07/2006 19:52:03,entry=HSCL0406 Multiple partitions enabled.
```

lsvg command

Purpose

Displays information about volume groups.

Syntax

```
lsvg [-map | -lv | -pv ] [ -field FieldName ] [ -fmt Delimiter ] VolumeGroup...
```

Description

The **lsvg** command displays information about volume groups. If you use the *VolumeGroup* parameter, only the information for that volume group is displayed. If you do not use the *VolumeGroup* parameter, a list of the names of all defined volume groups is displayed.

When information from the Device Configuration database is unavailable, some of the fields will contain a question mark (?) in place of the missing data. The **lsvg** command attempts to obtain as much information as possible from the description area when the command is given a logical volume identifier.

Full scripting support is provided to the **lsvg** command by using the **-field** *FieldNames* and **-fmt** *Delimiter* flags. The **-field** flag will allow the user to select which output fields to display and in what order, while the **-fmt** flag provides scriptable output. The output fields will be displayed in the order they appear on the command line.

If you do not specify any flags, the following information will be displayed:

Volume group	Name of the volume group. Volume group names must be unique and can range from 1 to 15 characters.
Volume group state	State of the volume group. If the volume group is active, the state is either active/complete (indicating all physical volumes are active) or active/partial (indicating some physical volumes are not active). If the volume group is not active, the state is inactive.
Permission	Access permission: read-only or read-write.
Max LVs	Maximum number of logical volumes allowed in the volume group.
LVs	Number of logical volumes currently in the volume group.
Open LVs	Number of logical volumes within the volume group that are currently open.
Total PVs	Total number of physical volumes within the volume group.
Active PVs	Number of physical volumes that are currently active.
VG identifier	The volume group identifier.
PP size	Size of each physical partition.
Total PPs	Total number of physical partitions within the volume group.
Free PPs	Number of physical partitions not allocated.
Alloc PPs	Number of physical partitions currently allocated to logical volumes.
Quorum	Number of physical volumes needed for a majority.
VGDS	Number of volume group descriptor areas within the volume group.
Auto-on	Automatic activation at IPL (yes or no).
Concurrent	States whether the volume group is Concurrent Capable or Non-Concurrent Capable.
Auto-Concurrent	States whether you should auto activate the Concurrent Capable volume group in concurrent or non-concurrent mode. For volume groups that are Non-Concurrent Capable, this value defaults to Disabled.
VG Mode	The mode of the volume group: Concurrent or Non-Concurrent.
Node ID	Node id of this node if volume group is in concurrent node.
Active Nodes	Node ids of other concurrent nodes that have this volume group active.
Max PPs Per PV	Maximum number of physical partitions per physical volume allowed for this volume group.
Max PVs	Maximum number of physical volumes allowed in this volume group.
LTG size	Logical track group size, in number of kilobytes, of the volume group.
BB POLICY	Bad block relocation policy of the volume group.
SNAPSHOT VG	Snapshot volume group name if the snapshot volume group is active else snapshot volume group identifier.
PRIMARY VG	Original volume group name of a snapshot volume group if the original volume group is active else original volume group identifier.

Flags

-field

Specifies the list of fields to display. The following fields are supported if no flags are specified:

vgname	Name of the volume group. Volume group names must be unique systemwide and can range from 1 to 15 characters.
vgstate	State of the volume group. If the volume group is activated with the activatevg command, the state is either active/complete (indicating all physical volumes are active) or active/partial (indicating all physical volumes are not active). If the volume group is de-activated with the deactivatevg command, the state is inactive .
access	Access permission: read-only or read-write.
maxlvs	Maximum number of logical volumes allowed in the volume group.
numlvs	Number of logical volumes currently in the volume group.
openlvs	Number of logical volumes within the volume group that are currently open.
totalpvs	Total number of physical volumes within the volume group.
stalepvs	Number of PVs which are not current. The data is stale.
stalepps	Number of PPs which are not current. The data is stale.
totalpps	Total number of physical partitions within the volume group.
freepps	Number of physical partitions not allocated.
usedpps	Number of physical partitions currently allocated to logical volumes.
quorum	Number of physical volumes needed for a majority.
vgds	Number of volume group descriptor areas within the volume group.
auton	Automatic activation at IPL (yes or no).
pppervg	Maximum number of physical partitions allowed in this volume group.
ppperpv	Maximum number of physical partitions per physical volume allowed for this volume group.
maxpvs	Maximum number of physical volumes allowed in this volume group. This information is displayed only for 32 and 128 PV volume groups.
ltgsize	Logical track group size of the volume group. The maximum amount of data that can be transferred in one I/O request to the disks of the volume group. The LTG size will be displayed in kilobytes unless the LTG size is greater than 1 MB, in which case megabytes will be used. It is capable of dynamically determining the LTG size based-on the disk topology and it is listed as Dynamic . If that capability is disabled by the user with the option, then it will be listed as Static .
bbpolicy	Bad block relocation policy of the volume group.
hotspare	
autosync	

The following fields are supported if the **-lv** flag is specified:

lvname A logical volume within the volume group.

type Logical volume type.

lps Number of logical partitions in the logical volume.

pvs Number of physical partitions used by the logical volume.

lvstate State of the logical volume. `Opened/stale` indicates the logical volume is open but contains partitions that are not current. `Opened/syncd` indicates the logical volume is open and synchronized. `Closed` indicates the logical volume has not been opened.

mount File system mount point for the logical volume, if applicable.

The following fields are supported if the **-pv** flag is specified:

pvname
A physical volume within the volume group.

pvstate State of the physical volume.

totalpps
Number of physical partitions on the physical volume.

freepps
Number of free physical partitions on the physical volume.

dist The number of physical partitions allocated within each section of the physical volume: outer edge, outer middle, center, inner middle, and inner edge of the physical volume.

-fmt Specifies a delimiter character to separate output fields.

-pv Lists the following information for each physical volume within the group specified by the *VolumeGroup* parameter:

Physical volume
A physical volume within the group.

PVstate
State of the physical volume.

Total PPs
Total number of physical partitions on the physical volume.

Free PPs
Number of free physical partitions on the physical volume.

Distribution
The number of physical partitions allocated within each section of the physical volume: outer edge, outer middle, center, inner middle, and inner edge of the physical volume.

-lv Lists the following information for each logical volume within the group specified by the *VolumeGroup* parameter:

LV A logical volume within the volume group.

Type Logical volume type.

LPs Number of logical partitions in the logical volume.

PPs Number of physical partitions used by the logical volume.

PVs Number of physical volumes used by the logical volume.

Logical volume state

State of the logical volume. `Opened/stale` indicates the logical volume is open but contains partitions that are not current. `Opened/syncd` indicates the logical volume is open and synchronized. `Closed` indicates the logical volume has not been opened.

Mount Point

File system mount point for the logical volume, if applicable.

-map

Lists the following fields for each logical volume on the physical volume:

PVname:PPnum [LVname:LPnum [:Copynum] [PPstate]]

PVname

Name of the physical volume as specified by the system.

PPnum Physical partition number. Physical partition numbers can range from 1 to 1016.

LVname

Name of the logical volume to which the physical partitions are allocated. Logical volume names must be system-wide unique names, and can range from 1 to 64 characters.

LPnum Logical partition number. Logical partition numbers can range from 1 to 64,000.

Copynum

Mirror number.

PPstate Only the physical partitions on the physical volume that are not current are shown as stale.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

1. To display the names of all volume groups within the system, type:

```
lsvg
```

2. To display information about volume group **vg02**, type:

```
lsvg vg02
```

The characteristics and status of both the logical and physical partitions of volume group **vg02** are displayed.

3. To display the names, characteristics, and status of all the logical volumes in volume group **vg02**, type:

```
lsvg -lv vg02
```

Related Information

The **mkvg** command, the **chvg** command, the **extendvg** command, the **reducevg** command, the **mirrorios** command, the **unmirrorios** command, the **activatevg** command, the **deactivatevg** command, the **importvg** command, the **exportvg** command, and the **syncvg** command.

lsvopt command

Purpose

Lists and displays information about the systems virtual optical devices.

Syntax

```
lsvopt [ -vtd VirtualTargetDevice] [-field Fieldnames] [-fmt Delimiter]
```

Description

The **lsvopt** command displays information about file backed virtual optical devices in the Virtual I/O Server. If no flags are specified, a list of all defined file backed virtual optical devices, loaded media, and the size of the media is displayed. If the **-vtd** flag is specified, the media loaded and its size is displayed for the given virtual target device.

This command will provide full scripting support through the use of the **-field** and **-fmt** flags.

Flags

-vtd	Specifies the Virtual target device of the file backed virtual optical media
-field <i>FieldName</i>	The following fields are supported if no flags are specified: vtd Virtual target device of file backed virtual optical device media File name of loaded media or No Media size Size of loaded media or n/a
-fmt <i>Delimiter</i>	Specifies a delimiter character to separate output fields.

Examples

To display information about all file backed virtual optical devices in the Virtual I/O Server, type the following command:

```
lsvopt
```

The system displays output similar to the following:

VTD	Media	Size(mb)
testopt	No Media	n/a
vtopt2	No Media	n/a
vtopt3	clientCD	640
vtopt4	No Media	n/a
vtopt5	No Media	n/a
vtopt6	No Media	n/a
vtopt7	No Media	n/a
vtopt8	No Media	n/a
vtopt9	No Media	n/a
vtopt10	No Media	n/a
vtopt11	No Media	n/a
vtopt12	No Media	n/a
vtopt13	clientCD	640
vtopt14	No Media	n/a
vtopt15	No Media	n/a
vtopt16	installDVD1	1000
vtopt17	installDVD2	100

man command

Purpose

Displays manual entries online.

Syntax

`man Command`

Description

The **man** command provides reference information on commands specified by name.

Exit Status

This command returns the following exit values:

0 Successful completion.
>0 An error occurred.

Examples

1. To display information about the **grep** command, enter:

```
man grep
```

Related Information

The **more** command.

migratepv command

Purpose

Moves allocated physical partitions from one physical volume to one or more other physical volumes.

Syntax

`migratepv [-lv LogicalVolume] SourcePhysicalVolume DestinationPhysicalVolume ...`

Description

The **migratepv** command moves allocated physical partitions and the data they contain from the *SourcePhysicalVolume* to one or more other physical volumes, *DestinationPhysicalVolume*. All physical volumes must be within the same volume group. The specified source physical volume cannot be included in the list of *DestinationPhysicalVolume* parameters.

The allocation of the new physical partitions follows the policies defined for the logical volumes that contain the physical partitions being moved.

If you specify a logical volume that contains the boot image, the **migratepv -lv** command attempts to find enough contiguous partitions on one of the target physical volumes. If the migration is successful, the **migratepv** command will indicate a change in the boot device as well as the new boot physical volume. The attempted migration fails if the **migratepv -lv** command is unable to find enough contiguous space to satisfy the request.

Note: All logical volume manager migration functions work by creating a mirror of the logical volumes involved, then resynchronizing the logical volumes. The original logical volume is then removed. If the **migratepv** command is used to move a logical volume containing the primary dump device, the system will not have an accessible primary dump device during the execution of the command. Therefore, a dump taken during this execution may fail.

Flags

-lv Moves only the physical partitions allocated to the specified logical volume and located on the specified source physical volume.

Exit Status

Return code	Description
8	The physical volume is not assigned to a volume group

Examples

1. To move physical partitions from **hdisk1** to **hdisk6** and **hdisk7**, type:

```
migratepv hdisk1 hdisk6 hdisk7
```

Physical partitions are moved from one physical volume to two others within the same volume group.

2. To move physical partitions in logical volume **lv02** from **hdisk1** to **hdisk6**, type:

```
migratepv -lv lv02 hdisk1 hdisk6
```

Only those physical partitions contained in **lv02** are moved from one physical volume to another.

Related Information

The **lspv** command.

IVM migrpar command

Purpose

Moves active or inactive logical partitions from one physical system to another physical system.

Syntax

To validate a migration

```
migrpar [-o v -m <managed system> -t <managed system> --ip <target HMC/IVM IP address> [-u <target HMC/IVM username>]] -p <partition name> | --id <partition ID> [-n <profile name>] [-f <input data file> | -i "<input data>"] [-w <wait time>] [-d <detail level>]
```

To migrate a logical partition

```
migrpar [-o m -m <managed system> -t <managed system> --ip <target HMC/IVM IP address> [-u <target HMC/IVM username>]] -p <partition name> | --id <partition ID> [-n <profile name>] [-f <input data file> | -i "<input data>"] [-w <wait time>] [-d <detail level>] [--async] [-v] | [ -redundantpgvios { 0 | 1 | 2 }]
```

To stop a migration

```
migrpar [-o s -m <managed system> {-p <partition name> | --id <partition ID>} [--help]
```


To recover from a failed partition migration

```
migrpar [-o r -m <managed system> [--ip <target HMC/IVM IP address>] [-u <target HMC/IVM username>] ]  
[-p <partition name> | --id <partition ID> ][--force] [--help]
```

Description

The **migrpar** command validates, starts, stops, and recovers a partition migration. The Integrated Virtualization Manager determines which type of migration to perform based on the state of the partition referenced in the command.

Flags

-o	The partition migration operation. Possible values: <ul style="list-style-type: none">• s - stops a partition migration• m - validates and migrates a partition if the validation succeeds• r - recover from a failed partition migration• v - validates a partition migration
-m <i>managed system</i>	The name of the source managed system for the partition migration operation. The name may be the user-defined name for the managed system, or may be in the form <i>ttt-mmm*sssssss</i> , where <i>ttt</i> is the machine type, <i>mmm</i> is the model, and <i>sssssss</i> is the serial number of the managed system.
-t <i>managed system</i>	The name of the target system for the partition migration operation. The name may be the user-defined name for the managed system, or may be in the form <i>tttmmm*sssssss</i> , where <i>ttt</i> is the machine type, <i>mmm</i> is the model, and <i>sssssss</i> is the serial number of the managed system. The -t flag is required when using the -o m or -o v flags.
--ip <i>target HMC/IVM IP address</i>	The IP address or hostname of the target system managing the Hardware Management Console or Integrated Virtualization Manager.
-u <i>target HMC/IVM username</i>	The user name to use on the Integrated Virtualization Manager that manages the target system or the HMC. If --ip flag is specified and -u flag is not specified, the user name on the source HMC or Integrated Virtualization Manager is used on the target HMC or Integrated Virtualization Manager.
-p <i>partition name</i>	The name of the partition on which to perform the migration.
--id <i>partition ID</i>	The ID of the partition on which to perform the migration.
-n <i>profile name</i>	The name of the partition profile to be created for the migrated partition on the destination managed system. If this option is omitted when migrating a partition, then the last activated profile for the partition will be replaced with the current partition configuration on the destination managed system. This option is only valid when migrating a partition or validating a partition migration. Note: This option is only valid if you are migrating to a system managed by the HMC. If you are migrating to a system managed by the Integrated Virtualization Manager, the option is not used.

-f *input data file*

The name of the file containing the input data for the **migr1par** command. The data given in the file specified with the **-f** flag, or the data specified with **-i**, must be in comma-separated (CSV) format. These switches can be used with the migrate (**-o m**) and the validate (**-o v**) operations. The following attributes are supported:

```
virtual_scsi_mappings, virtual_fc_mappings, source_msp_name,  
source_msp_ipaddr, source_msp_id,  
dest_msp_name, dest_msp_ipaddr,  
dest_msp_id, shared_proc_pool_id,  
shared_proc_pool_name, paging_device  
primary_paging_vios_id  
primary_paging_vios_name
```

The data specified with the `virtual_scsi_mappings` or `virtual_fc_mappings` attribute consists of one or more source virtual SCSI or virtual fibre channel adapters to destination Virtual I/O Server logical partitions in the following format:

```
client_virtual_slot_num/dest_vios_lpar_name/  
dest_vios_lpar_id
```

Attribute names

dest_msp_id

Specifies the partition ID to use on the destination managed system.

dest_msp_ipaddr

Specifies the IP address of the mover service partition of the destination managed system.

Note: This value is verified by comparing it with the output of the **lstcpip -interfaces** command.

dest_msp_name

Specifies the mover service partition name of the destination managed system.

paging_device

The paging space device to use if a memory pool is used. A paging device is a block storage device that has been added to the memory pool and is not designated as a paging device for any other logical partition. If the `paging_device` value is a blank string, there is no paging device currently assigned.

primary_paging_vios_id

The ID of the primary paging Virtual I/O Server (VIOS) partition that provides access to the paging space devices for the shared memory partitions. A paging VIOS partition is a VIOS logical partition that is assigned to the shared memory pool.

primary_paging_vios_name

The name of the primary paging VIOS partition that provides access to the paging space devices for the shared memory partitions. A paging VIOS partition is a VIOS logical partition that is assigned to the shared memory pool.

shared_proc_pool_id

The unique decimal identifier for a shared processing pool in which this logical partition should be on the target system. The default ID is zero. If Integrated Virtualization Manager is the destination managed system, the ID must equal 0.

shared_proc_pool_name

Specifies the name of the shared processor pool in which this partition should be on the target system. This attribute is only valid for logical partitions using shared processors. The default value is `DefaultPool`.

source_msp_id
Specifies the partition ID to use on the source managed system. On IVM, this ID must equal the ID of the Virtual I/O Server.

source_msp_ipaddr
Specifies the IP address of the mover service partition of the source managed system.
Note: This value is verified by comparing it with the output of the **lstcpip -interfaces** command.

source_msp_name
Specifies the mover service partition name of the source managed system. On IVM, this name must equal the name of the Virtual I/O Server logical partition.

virtual_fc_mappings
Comma-separated list of virtual fibre channel adapters. Each item in this list has the format `slot_num/vios_lpar_name/vios_lpar_id`. For example, `4/vios2/3` specifies a virtual fibre channel adapter on a client logical partition with a virtual slot number of 4, a VIOS partition name of `vios2`, and the ID of the destination VIOS logical partition of 3.

virtual_scsi_mappings
Comma-separated list of virtual SCSI adapters. Each item in this list has the format `slot_num/vios_lpar_name/vios_lpar_id`. For example, `2/vios/1` specifies a client virtual SCSI adapter with a virtual slot number of 2, a VIOS partition name of `vios`, and the ID of the destination VIOS logical partition of 1.

-i input data
The input data for the **migr_lpar** command. The format of the filter data is `attr_name1=value,attr_name2=value,...` or `attr_name1=value1,value2,...`

-w wait time
The maximum time, in minutes, to wait for the operating system commands issued by the HMC or Integrated Virtualization Manager to the partition to be migrated to complete.

-d detail level
The level of detail requested from the operating system commands issued by the HMC or Integrated Virtualization Manager to all partitions participating in the migration. Values range from 0 (none) to 5 (highest).

--async
The command returns after the validation is completed. This flag does not wait for the migration to complete. This is only valid with the **-o m** flag.

-v
Enables verbose mode for the partition migration operation. When verbose mode is enabled, detail messages and warning messages are displayed for a successful partition migration. Detail messages and warning messages are always displayed for a partition migration that fails, regardless of whether this option is specified.

--force
Forces a recover operation to proceed when errors are encountered. This option is only valid when recovering from a failed partition migration and can be initiated from either the source VIOS logical partition or the destination managed system. If the operation is initiated from the destination managed system, only the destination managed system is recovered.

redundantpgvios {0 | 1 | 2}
Specifies whether to configure the partition for redundancy on the target managed system. Possible values follow:

- 0** Do not configure the partition for redundancy on the target managed system.
- 1** Configure the partition for redundancy on the target managed system.
- 2** Configure the partition for redundancy if possible on the target managed system. If redundancy is not possible, configure as nonredundant.

If you do not specify the parameter, the current redundancy level of the migrating partition is used.

--help
Displays the help text for this command and exit.

Exit Status

This command has a return code of 0 on success.

Examples

1. To validate a partition migration, type the following command:

```
migr1par -o v -m migfspL1 --ip migivm2 -t migfspL2 --id 5  
-i "shared_proc_pool_name=ProcPoolA"
```

2. To perform a partition migration, type the following command:

```
migr1par -o m -m migfspL1 --ip migivm2 -t migfspL2 --id 5 -i "source_msp_id=1,  
source_msp_ipaddr=9.3.252.192,dest_msp_id=1,dest_msp_ipaddr=9.3.126.77"
```

3. To perform a partition migration for logical partitions that use virtual fibre channel adapters, type the following command:

```
migr1par -o m -m migfspL1 --ip migivm2 -t migfspL2 --id 5  
-i "virtual_fc_mappings=5/VIOS/1,6/VIOS3/3"
```

4. To stop a partition migration, type the following command:

```
migr1par -o s -m migfspL1 --id 5
```

5. To recover from a failed partition migration, type the following command:

```
migr1par -o r -m migfspL1 --id 5
```

mirrorios command

Purpose

Mirrors all the logical volumes on rootvg. This command reboots the partition upon completion.

Syntax

```
mirrorios [ -f ][ -defer ] [ PhysicalVolume ...]
```

Description

The **mirrorios** command takes all the logical volumes on the rootvg volume group and mirrors those logical volumes. The target physical drives must already be members of the volume group.

The **mirrorios** command attempts to mirror the logical volumes onto any of the disks in a volume group. To control which drives are used for mirroring, you must include the list of disks in the input parameters, *PhysicalVolume*. Mirror strictness is enforced. The **mirrorios** command mirrors the logical volumes, using the default settings of the logical volume being mirrored.

The Virtual I/O Server will restart when the command is completed and you are prompted to continue. If you specify the **-f** option, the command runs without prompting you.

Note: For best results, mirror the rootvg volume group on all Virtual I/O Server partitions.

Only the prime administrator (padmin) can run this command.

Flags

-f	Runs the command without prompting you to continue.
-defer	Specifies that you do not want to receive a query about a system restart. The system should be restarted later.
<i>PhysicalVolume</i>	Specifies the target physical volume name. The volume must already be a member of the volume group.

Exit Status

5	The Virtual I/O Server is already mirrored
6	Boot LV not found
7	The physical volume appears to belong to another volume group

Examples

1. To mirror the Virtual I/O Server root volume group to physical volume **hdisk4**, type the following command:

```
mirrorios -force
hdisk4
```

2. To mirror the Virtual I/O Server root volume group to physical volume **hdisk4**, but defer a system reboot, type the following command:

```
mirrorios -defer hdisk4
```

Related Information

The `activatevg` command, the `chvg` command, the `deactivatevg` command, the `exportvg` command, the `importvg` command, the `lsvg` command, the `mkvg` command, the `syncvg` command, the `unmirrorios` command, and the `alt_root_vg`.

IVM mkauthkeys command

Purpose

Allows key-based SSH authentication between two systems. This updates the `~/.ssh/authorized_keys2` file with the specified public key. It can also be used to push the users public key to a remote Integrated Virtualization Manager or HMC system.

Syntax

To add the SSH key as an authorized key locally:

```
mkauthkeys { -a | --add } <key string>
```

To remove the SSH key locally:

```
mkauthkeys { -r | --remove } [ -u <user> ] <key string>
```

To exchange public keys with a remote system:

```
mkauthkeys { -a | --add } -- ip <remote system> [ -u <user> ] <key string>
```

To test the remote non-interactive authentication using the SSH key:

```
mkauthkeys --test -- ip <remote system> [ -u <user> ]
```

Description

The `mkauthkeys` command updates the Integrated Virtualization Manager user's `authorized_keys2` file.

Flags

-a	Adds the ssh command key.
-g	Displays the public key for the specified user and generates the user's public and private key pair if they do not exist.
-r	Removes the key for the specified user ID and host.
--add	Adds the ssh command key.
--remove	Removes the key for the specified user ID and host.
--test	Verifies authentication to the remote host
--ip <remote server IP>	Allows installing this user's public key on the specified remote HMC or Integrated Virtualization Manager system for the user specified with -u flag. If -u flag is not specified, the remote user's public key will be installed on the local system.
-u <i>username</i>	Specifies the user name to add or remove the key. You must have hmcsuperadmin or PAdmin authority to add or remove key for other users.
<i>key string</i>	The ssh command key to add to or the ID to remove.

Exit Status

This command has a return code of 0 on success.

Examples

1. To add the SSH key generated for user *joe@somehost* , type the following command:
`mkauthkeys -a 'adB8fqeZs2d-gg+q joe@somehost'`
 2. To display the current user's public key, type the following command:
`mkauthkeys -g`
 3. To display the public key for *fred*, type the following command:
`mkauthkeys -g -u fred`
 4. To remove the SSH key generated for user *joe@somehost*, type the following command:
`mkauthkeys -r 'adB8fqeZs2d-gg+q joe@somehost'`
 5. To remove all SSH keys generated for user *joe@somehost*, type the following command:
`mkauthkeys -r 'joe@somehost'`
 6. To add the SSH key generated for this user to a remote.host as user *fred*, type the following command:
`mkauthkeys -a --ip remote.host -u fred`
- Note:** The user will be prompted for the password on the remote.host.
7. To enable *user* from *somesystem* with public key *ssh-rsa thersakeygoeshere=* to access the Virtual I/O Server without using a password. type the following command:
`mkauthkeys -a ssh-rsa thersakeygoeshere= user@somesystem`
 8. To remove the key from the authorized key list, type the following command:
`mkauthkeys -r ssh-rsa thersakeygoeshere= user@somesystem`
 9. To remove all keys that end with the string *user@somesystem*, type the following command:
`mkauthkeys -r user@somesystem`
 10. To allow the padmin user to remove a key for any user, type the following command:
`mkauthkeys -r -u user ssh-rsa thersakeygoeshere= user@somesystem`
 11. To add the current user's local public key to the authorized keys list on a remote system, and adds the user's remote public key to the authorized keys list on the local system, type the following command:
`mkauthkeys -a --ip othersystem.com`

12. To add the current user's local public key to the authorized keys list for remote user *user* on a remote system, and adds the remote public key of *user* to the authorized keys list for the current user on the local system, type the following command:

```
mkauthkeys -a --ip othersystem.com -u user
```

13. To verify non-interactive authentication using the SSH key for the current user, type the following command:

```
mkauthkeys --test --ip othersystem.com
```

Note: If it returns 0, then non-interactive authentication is working properly. If **mkauthkeys** returns nonzero, then non-interactive authentication is not configured properly and displays the following message: [VIOSE0104200B-0217] Permission denied (publickey,password,keyboard-interactive).

14. To verify non-interactive authentication using the SSH key for the current user on local system to specify the user on a remote system, type the following command:

```
mkauthkeys --test --ip othersystem.com -u user
```

mkbdsp command

Purpose

Assign storage from a storage pool to be a backing device for a virtual SCSI adapter.

Syntax

Create a backing device file or logical volume:

```
mkbdsp [-sp StoragePool] Size -bd BackingDevice
```

Assign an existing file or logical volume as a backing device:

```
mkbdsp [-sp StoragePool]-bd BackingDevice -vadapter ServerVirtualSCSIAdapter -tn TargetDeviceName
```

Create a new file or logical volume as a backing device:

```
mkbdsp [-sp StoragePool] Size [-bd BackingDevice] -vadapter ServerVirtualSCSIAdapter -tn TargetDeviceName
```

Description

The **mkbdsp** command assigns a backing device to a virtual SCSI server adapter. If the **-sp** flag is not specified the default storage pool will be used. The storage pool must be specified when working with file backed devices. The default storage pool will still be used when working with logical volumes. If the storage size is given, the **mkvdev** command will create a backing device of at least the specified size and assign it as the backing device. When working with file backed devices the **-bd** flag must be specified. The system will not generate a name. The type of backing device created will be determined by the storage pool type. Size can be given in megabytes (###M/n) or gigabytes (###G/g) or physical partition (###).

Note:

- Specifying physical partitions only works for logical volume backing devices.
- The backing device specified cannot be assigned to a shared memory pool (to be used as a paging space device by a shared memory partition).

You must specify the name for the newly created backing device using the **-bd** flag in combination with the size parameter. Naming the backing device is optional when working with logical volumes. You also have the option of assigning the name for the newly created virtual target device using the **-tn** flag in combination with the **-vadapter** flag.

Flags

-bd <i>BackingDevice</i>	Specifies the backing device.
-sp <i>StoragePool</i>	Specifies the storage pool to be used.
-vadapter	Specifies the virtual SCSI server adapter.
-tn	Specifies the name of the target device.

Note: The only values accepted are alphanumeric, a dash, an underscore, or a period.

Exit Status

23	Specified storage pool is not a valid storage pool.
26	Specified name is already used. Choose a different name.
34	Specified name is reserved. Choose a different name.

Examples

To create a virtual target device that maps a 3 GB backing device from the default storage pool the virtual SCSI server adapter vhost3, type the following command:

```
mkbdsp -bd bname 3g -vadapter vhost3
```

mkdir command

Purpose

Creates one or more new directories.

Syntax

```
mkdir [ -m Mode ] [ -p ] Directory ...
```

Description

The **mkdir** command creates one or more new directories specified by the *Directory* parameter. Each new directory contains the standard entries **.** (dot) and **..** (dot-dot). You can specify the permissions for the new directories with the **-m** *Mode* flag.

The owner-ID and group-ID of the new directories are set to the process's effective user-ID and group-ID, respectively. The setgid bit setting is inherited from the parent directory. To change the setgid bit, you can either specify the **-m** *Mode* flag or issue the **chmod** command after the creation of the directory.

Note: To make a new directory you must have write permission in the parent directory.

Flags

-m <i>Mode</i>	Sets the permission bits for the newly-created directories to the value specified by the <i>Mode</i> variable. The <i>Mode</i> variable takes the same values as the <i>Mode</i> parameter for the chmod command, either in symbolic or numeric form.
-----------------------	--

When you specify the **-m** flag using symbolic format, the op characters **+** (plus) and **-** (minus) are interpreted relative to the assumed permission setting **a=rwx**. The **+** adds permissions to the default mode, and the **-** deletes permissions from the default mode. Refer to the **chmod** command for a complete description of permission bits and formats.

-p Creates missing intermediate path name directories. If the **-p** flag is not specified, the parent directory of each-newly created directory must already exist.

Intermediate directories are created through the automatic invocation of the following **mkdir** commands:

```
mkdir -p -m $(umask -S),u+wx $(dirname Directory) &&  
mkdir [-m Mode] Directory
```

where the [-m Mode] represents any option supplied with your original invocation of the **mkdir** command.

The **mkdir** command ignores any Directory parameter that names an existing directory. No error is issued.

Exit Status

This command returns the following exit values:

- 0 All the specified directories were created successfully, or the **-p** option was specified and all the specified directories now exist.
- >0 An error occurred.

Examples

1. To create a new directory called Test in the current working directory, enter:

```
mkdir Test
```

The Test directory is created with default permissions.

2. To create a new directory called Test with rwxr-xr-x permissions in the previously created /home/demo/sub1 directory, enter:

```
mkdir -m 755 /home/demo/sub1/Test
```

3. To create a new directory called Test with default permissions in the /home/demo/sub2 directory, enter:

```
mkdir -p /home/demo/sub2/Test
```

The **-p** flag creates the /home, /home/demo, and /home/demo/sub2 directories if they do not already exist.

Files

`/usr/bin/mkdir` Contains the **mkdir** command.

Related Information

The **chmod** command, **rm** command.

IVM mkgencfg command

Purpose

Performs the initial logical partition configuration for the managed system. This command is valid only in an Integrated Virtualization Manager environment.

Syntax

```
mkgencfg -o init [-i "ConfigurationData" ] [ -m ManagedSystem ]
```

Description

The **mkgencfg** command performs the initial logical partition configuration for the managed system. As part of the initial configuration, virtual Ethernet adapters are created in the management partition. The virtual Ethernet MAC address prefix can optionally be configured through this command.

Flags

-o *Operation*

The type of operations:

-i *ConfigurationData*

init - Perform initial logical partition configuration actions for the managed system
The configuration data consists of attribute name/value pairs, which are in a comma separated value (CSV) format. The format of a configuration record is as follows:
"attribute-name=value,attribute-name=value,..."

Note that certain attributes accept a comma separated list of values, as follows:

"attribute-name=value,value,...",..."

When you specify a list of values, the attribute name/value pair must be enclosed in double quotation marks. Depending on the shell being used, nested double quotation marks might need to be preceded by an escape character.

Valid Attributes for Configuration Data:

mac_prefix

The prefix needs to be a 3 byte hexadecimal value. It specifies the first 2.5 bytes of the MAC address to be assigned to all virtual Ethernet adapters for this managed system. The value cannot be a multicast address (010000 bit must be off), and needs to be a private address (020000 bit must be on). For example, a valid MAC address prefix is 0642A0.

pend_configured_max_lpars

The maximum number of partitions supported by the management partition after the next restart.

virtual_eth_mac_base_value

The virtual Ethernet MAC address base value is a per partition value. The base value is used to build the MAC addresses for each virtual Ethernet adapter in that partition. The base value is the first 5 bytes of the MAC address. The virtual slot number of the virtual Ethernet adapter makes up the last byte. If you do not assign a base value, one is generated automatically using the following format: Base value = 0xSSSSSBBBBB

Where SSSSS is the system wide MAC address prefix andBBBBB is a randomly generated sequence of bits (guaranteed to be unique within this physical system).

Note:

1. The system wide MAC prefix is randomly generated unless you override it using `mkgencfg -o init -i mac_prefix`.
2. If the base value is automatically generated, then the MAC address format for a virtual Ethernet adapter is 0xSSSSSBBBBBNN, where NN is the slot number. If you specify the base value using **mkgencfg** for partition 1 or **mksyscfg/chsyscfg** for any other partition, then the format is 0xBBBBBBBBBBNN, whereBBBBBBBBB is the base value that you specified.
3. If a virtual Ethernet adapter is in a slot greater than or equal to 256, then the slot number overflows into the base value, since it does not fit in 1 byte. For example, Integrated Virtualization Manager handles it as 0xBBBBBBBBBB00 + 0x00000000NNNN.
4. The **mkgencfg** command is the only way to specify the base value for partition 1. After this value is set, you cannot change it without losing all partition configuration. If you need to change the value, use the **lpcfgop** command. The value is set implicitly when **mkgencfg** is running. The value will be generated automatically, unless you specify the value. The **mkgencfg** command runs implicitly when you run your first **change** command. You can change the base value for any other partition, when the partition is powered off.

A side effect of setting the base value for partition 1 with **mkgencfg**, is that the system wide prefix is set to the first 2.5 bytes of the partition 1 base value. As a result, there are restrictions on setting both `mac_prefix` and `virtual_eth_mac_base_value` with **mkgencfg** at the same time. If you specify both, the `mac_prefix` value needs to be equal to the first 2.5 bytes of the `virtual_eth_mac_base_value` value.

-m ManagedSystem

The name of the managed system. This attribute is optional because there is only one system to manage. The name might either be the user-defined name for the managed system, or be in the form `ttt-mmm*sssssss`, where `ttt` is the machine type, `mmm` is the model, and `sssssss` is the serial number of the managed system.

Exit Status

This command has a return code of zero on success.

Security

This command is not accessible by users with the ViewOnly role.

Examples

1. To initialize the logical partition configuration for the managed system using defaults, type:
`mkgencfg -o init`
2. To initialize the logical partition configuration for the managed system with support for 17 partitions and a MAC prefix of 0x06ABC0, type:

```
mkgencfg -o init -i "pend_lpm_max_lpars=17,mac_prefix=06ABC0"
```

mkkrb5clnt command

Purpose

Configures a Kerberos client.

Syntax

```
mkkrb5clnt -h | -r <realm> { -c <KDC> -s <server> | -l {ldapservice | ldapservice:port} [-c <KDC> -s<server>] } [-a<admin> ] -d <domain> [-A] [-i <database>] [-K] [-T] | -i <database> | -U [-a <admin>]
```

Description

The `mkkrb5clnt` command configures a Kerberos client.

Flags

<code>-h</code>	Specifies that the command will only display the valid command syntax
<code>-c KDC</code>	Specifies the KDC server
<code>-r realm</code>	Specifies the full realm name for which the Kerberos client is to be configured.
<code>-s server</code>	Specifies the fully qualified host name for Kerberos admin server
<code>-U</code>	Undo the setup from the previous configuration command
<code>-a admin</code>	Specifies the principal name of the Kerberos server admin
<code>-d domain</code>	Specifies the complete domain name for the Kerberos client
<code>-A</code>	Specifies root to be added as a Kerberos administrative user
<code>-i database</code>	Configures integrated Kerberos authentication
<code>-K</code>	Specifies Kerberos to be configured as the default authentication scheme
<code>-T</code>	Specifies the flag to acquire the server administration TGT based administration ticket.
<code>-l ldapservice ldapservice:port</code>	For clients, specifies the LDAP directory server to use for Administration server and KDC discovery using LDAP. If the <code>-l</code> flag is used, then the <code>-c</code> and <code>-s</code> flags are optional. If the <code>-l</code> option is not used, the <code>-c</code> and <code>-s</code> flags must be specified. The port number can optionally be specified.

Exit Status

Return code	Description
0	Success
1	Invalid flag, argument, or command failure

Examples

- To display the command syntax, type the following command:

```
mkkrb5clnt -h
```
- To configure `testbox.com` as the client, make root as the server admin, configure integrated login, configure Kerberos as default authentication scheme, type the following command:

```
mkkrb5clnt -c bob.kerberso.com -r KERBER.COM -s bob.kerberso.com -d testbox.com -A -i files -K -T
```

mkldap command

Purpose

Sets up Virtual I/O Server as a Lightweight Direct Access Protocol (LDAP) client.

Syntax

```
mkldap -host serverlist -bind bindDN -passwd bindpwd [ -base baseDN ] [ -port serverport ] [ -timeout cacheTimeout ] [ -csize cacheSize ] [ -threads NumberOfThreads ] [ -hbeatint heartBeatInt ] [-keypath SSL_database_path ] [-keypasswd SSL_password ] [ -auth authType ] [ -users userlist | ALL]
```

mkldap -deconfig

Description

The **mkldap** command is used to set up the Virtual I/O Server as an LDAP client. The server bind distinguished name (DN) and password for client access to the LDAP server. The **mkldap** command saves server bind DN, password, server name, the SSL key path and password, and other configuration attributes to the `/etc/security/ldap/ldap.cfg` file. The **mkldap** command saves the bind password and SSL key password (if SSL is configured) to the `/etc/security/ldap/ldap.cfg` file in encrypted format.

Note: These encrypted passwords are system specific, and can only be used by the `secldapclntd` daemon on the system where they are generated.

You can supply multiple LDAP servers to the **mkldap** command during client setup. In this case, the client contacts the servers in the supplied order and establishes connection to the first server to which the client can successfully bind.

The LDAP client communicates to the LDAP server through a client side daemon, **secldapclntd**.

The **secldapclntd** command is enabled or disabled by using the **startnetsvc** and **stopnetsvc** commands.

Flags

-host <i>serverlist</i>	Specifies comma-separated list of host names.
-bind <i>bindDN</i>	Specifies the DN (distinguished name) to bind to the LDAP server.
-passwd <i>bindpwd</i>	Specifies the clear text password for the bindDN used to bind to the LDAP server.
-base <i>baseDN</i>	Specifies the base DN for the mkldap command, in which to search for the user base DN and group base DN. If you do not specify this flag, the entire database is searched.
-port <i>serverport</i>	Specifies the port number to which the LDAP server is listening.
-timeout <i>cachetimeout</i>	Specifies the maximum time length that a cache entry expires. Set this value to 0 to disable caching.
-csize <i>cacheSize</i>	Specifies the maximum number of user entries used in the client side daemon cache.
-threads <i>NumberOfThreads</i>	Specifies the numbers of threads that the client side daemon uses.
-hbeatint <i>heartBeatInt</i>	Specifies the time interval of heartbeats between the client and the LDAP server.
-keypath <i>SSL_database_path</i>	Specifies the full path to the SSL database. Note: This requires the <code>ldap.max_crypto_client</code> fileset to be installed.
-keypasswd <i>SSL_password</i>	Specifies the password for the SSL key. Note: This requires the <code>ldap.max_crypto_client</code> fileset to be installed.
-auth <i>authType</i>	Specifies the authentication mechanism that is used to authenticated users. Valid values are <code>unix_auth</code> and <code>ldap_auth</code> .
-users <i>userlist</i>	Specifies the comma-separated list of user names to enable for LDAP authentication. Specify ALL to enable all users on the client.
-deconfig <i>bindpwd</i>	Specifies that the previous client setup to the LDAP client configuration file should be undone.

Exit Status

0 Success

Examples

Run one of the following **mkldap** commands with the **-u** flag for a user ID to become an LDAP user ID at client setup time:

- ```
mkldap -host ldapserv1 -bind cn=admin -passwd adminpwd -users user1,user2
mkldap -host ldapserv1 -bind cn=admin -passwd adminpwd
```
- To setup the client to talk to the server3.your\_company.com LDAP server using SSL, enter the following command:

```
mkldap -bind cn=admin -passwd adminpwd -host server3.your_company.com
-base o=mycompany,c=us -keypath /usr/ldap/clientkey.kdb
-keypasswd keypwd -users user1,user2
```

Either of these commands set up the local host as the client of the LDAP server running on host **ldapserv1** and **cn=admin** and **-passwd adminpwd** are the LDAP server administrator DN and password.

## Files

`/etc/security/ldap/ldap.cfg` Contains the **mkldap** command, and server bind DN, password, server name, the SSL key path and password, and other configuration attributes.

## Related Information

The **ldapadd** command and the **ldapsearch** command.

---

## mklv command

### Purpose

Creates a logical volume.

### Syntax

```
mklv [-mirror] [-lv NewLogicalVolume | -prefix Prefix] VolumeGroup Size [PhysicalVolume ...]
```

### Description

The **mklv** command creates a new logical volume within the *VolumeGroup*. If you specify one or more physical volumes with the *PhysicalVolume* parameter, only those physical volumes are available for allocating physical partitions; otherwise, all the physical volumes within the volume group are available.

The allocation policy is to use a minimum number of physical volumes.

The *Size* parameter specifies the minimum size the logical volume should be. When specifying *Size* the following conventions must be used:

| Size   | Minimum logical volume size |
|--------|-----------------------------|
| ###M/m | ### MB                      |
| ###G/g | ### GB                      |

## Flags

- lv** Specifies the logical volume name to use instead of using a system-generated name. Logical volume names must be unique system wide name, and can range from 1 to 15 characters.
- mirror** Activates mirroring for this logical volume.
- prefix** Specifies the Prefix to use instead of the prefix in a system-generated name for the new logical volume. The prefix must be less than or equal to 13 characters. The name cannot begin with a prefix already defined in the **PdDv** class in the Device Configuration Database for other devices, nor be a name already used by another device.

## Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

## Examples

1. To make a logical volume in volume group **vg02** with a minimum size of 1 Mb , type:  

```
mklv vg02 1M
```
2. To make a logical volume in **vg03** with 1GB chosen from physical volumes **hdisk5**, **hdisk6**, and **hdisk9**, type:  

```
mklv vg03 1G hdisk5 hdisk6 hdisk9
```
3. To request a logical volume with a minimum size of 10MB, type:  

```
mklv VGNAME 10m
```

where *VGNAME* is the name of your logical volume.

## Related Information

The **lslv** command, the **extendlv** command, and the **rmlv** command.

---

## mklvcopy command

### Purpose

Creates a mirror of a logical volume.

### Syntax

```
mklvcopy LogicalVolume [PhysicalVolume ...]
```

### Description

The **mklvcopy** command creates a mirror (an additional copy) of a *LogicalVolume*. The *LogicalVolume* parameter can be a logical volume name or logical volume ID. You can request that the new copy of the logical volume be allocated on specific physical volumes (within the volume group) with the *PhysicalVolume* parameter; otherwise, all the physical volumes within the volume group are available for allocation. The new copy of the logical volume will be placed on a separate physical volume.

**Note:** Only one additional copy of a logical volume can be created.

### Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

## Examples

1. To create a copy of the logical volume **lv01**, so that a total of two copies exist, type:  
`mk1vcopy lv01`

## Related Information

The `extendlv` command, the `lslv` command, the `mklv` command, the `rmlv` command, and the `rmlvcopy` command.

---

## mkpath command

### Purpose

Adds to the system another path to an MPIO capable device.

### Syntax

```
mkpath { [-dev Name] [-pdev Parent] [-conn Connection] } [-def]
```

### Description

The **mkpath** command defines, and possibly configures, one or more paths to the target device (**-dev *Name***). The paths are identified by a combination of the **-dev *Name***, **-pdev *Parent***, and **-conn *Connection*** flags. Both the target device and parent must be previously defined in the system to define a path. They both must be AVAILABLE to configure a path.

If the **-def *flag*** is specified, the **mkpath** command only defines the new path definition to the system. If the **-def** flag is not specified, the **mkpath** command attempts to define the path, if it does not already exist, before it attempts to configure the path. Configuring a path requires the path to already be defined and both the device and the parent device to already be configured.

The **mkpath** command displays a status message upon completion. It is possible for some paths to configure and others to fail.

Note that not all devices will be able to have paths manually defined by using the **mkpath** command. These limitations are due to the way that path information is stored for these devices. Fiber channel devices fall into this category.

The **mkpath** command provides status messages about the results of operation. Messages in one of the following formats will be generated:

#### **path [ available | defined ]**

This message is displayed when **mkpath** is run on a single path. If the path is successfully configured the message **path available** is displayed. If the path is not successfully configured and there is no explicit error code returned by the method, the message **path defined** is displayed.

#### **paths available**

This message is displayed if multiple paths were identified and all paths were successfully configured.

#### **some paths available**

This message is displayed if multiple paths were identified, but only some of them were successfully configured.

#### **no paths processed**

This message is generated if no paths were found matching the selection criteria.



## Flags

|                                |                                                                                                                                                                                                                                                                                                                   |
|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-conn</b> <i>Connection</i> | Indicates the connection information associated with the path to be added. This flag is required if the <b>-def</b> flag is specified.                                                                                                                                                                            |
| <b>-def</b>                    | Defines a new path to a device by adding a path definition to the system. The new path will not automatically be configured when the <b>-def</b> flag is specified. Note that only one path may be defined at a time. The <b>-conn</b> and the <b>-pdev</b> flags are required when the <b>-def</b> flag is used. |
| <b>-dev</b> <i>Name</i>        | Specifies the logical device name of the target device to which the path(s) are being added. The path(s) to be added are qualified by the <b>-pdev</b> and <b>-conn</b> flags.                                                                                                                                    |
| <b>-pdev</b> <i>Parent</i>     | Indicates the logical device name of the parent device associated with the path(s) to be added. This flag is required if the <b>-def</b> flag is specified.                                                                                                                                                       |

## Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

## Examples

1. To define and configure an already defined path between **scsi0** and the **hdisk1** device at **SCSI ID 5** and **LUN 0** (connection 5,0), enter:

```
mkpath -dev hdisk1 -pdev scsi0 -conn 5,0
```

The system displays a message similar to the following:

```
path available
```

2. To configure an already defined path from **fscsi0** to fiber channel disk **hdisk1**, type:

```
mkpath -dev hdisk1 -pdev fscsi0
```

The system displays a message similar to the following:

```
path available
```

3. To only add to the Customized Paths object class a path definition between **scsi0** and the **hdisk1** disk device at **SCSI ID 5** and **LUN 0**, enter:

```
mkpath -def -dev hdisk1 -pdev scsi0 -conn 5,0
```

The system displays a message similar to the following:

```
path defined
```

## Related Information

The `lspath` command and the `rmpath` command.

---

## mkrep command

### Purpose

Create the Virtual Media Repository.

### Syntax

```
mkrep -sp ParentStoragePool -sizeSize
```

## Description

The **mkrep** command creates the virtual media repository in the specified parent storage pool. The virtual media repository is used to store virtual optical media which can be conceptually inserted into file-backed virtual optical devices. See the **mkvdev** command for detail on how to create file-backed virtual optical devices.

The **-size** flag specifies the minimum size the repository should be. When specifying the *Size* the following conventions must be used:

| Size   | Minimum file storage pool size |
|--------|--------------------------------|
| ###M/m | ###MB                          |
| ###G/g | ###GB                          |

## Flags

**-size** *Size* Specifies the minimum size the repository should be.  
**-sp** *ParentStoragePool* Specifies the parent storage pool the repository should be created within. The parent storage pool must be a logical volume pool.

## Examples

To create the virtual media repository within logical volume storage pool *client\_data* with a size of at least 100 megabytes , type the following command:

```
mkrep -sp client_data -size 100m
```

---

## mksp command

### Purpose

Create a storage pool.

### Syntax

**Create a logical volume storage pool:**

```
mksp [-f] StoragePool PhysicalVolume ...
```

**Create a file storage pool:**

```
mksp -fb StoragePool -sp ParentStoragePool-size Size [-mirror]
```

### Description

The **mksp** command creates a new logical volume or file storage pool. Logical volume pools are used to store logical volume backing devices, file storage pools and the Virtual Media Repository. These pools are created using the physical volumes represented by the *PhysicalVolume* parameter.

If the system detects a description area from a volume group that is not varied on, it prompts you for confirmation in continuing with the command. The previous contents of the physical volume are lost, so you must be cautious when using the override function. By specifying the **-f** flag, you force the volume group to be created without sending a confirmation message.

File pools are used to store backing device files. A file pool is created within a logical volume pool, specified by the **-sp** ParentStorage Pool parameter.

The **-size** *Size* flag specifies the minimum size the pool should be. When specifying *Size* the following conventions must be used:

| Size   | Minimum file storage pool size |
|--------|--------------------------------|
| ###M/m | ###MB                          |
| ###G/g | ###GB                          |

## Flags

- f** Forces the storage pool to be created on the specified physical volume unless the physical volume is part of another storage pool or volume group in the Device Configuration Database or is an active volume group.
- In addition, the physical volume that you specify cannot be assigned to a shared memory pool (to be used as a paging space device by a shared memory partition).
- fb** *StoragePool* Specifies the name of the file storage pool to be created. The name must be a unique system wide name, and can range from 1 to 15 characters.
- mirror** Activates mirroring for this file storage pool.
- size** *Size* Specifies the minimum size the file storage pool should be.
- sp** *ParentStoragePool* Specifies the parent storage pool the file pool should be created within. The parent storage pool must be a logical volume pool

## Examples

1. To create a new logical volume storage pool from physical volumes hdisk3 and hdisk4 and with the name *client\_data*, type:

```
mksp -f client_data hdisk3 hdisk4
```

The new storage pool is created with the name *client\_data*.

2. To create a new file storage pool within logical volume storage pool *client\_data* with a size of at least 100 MB and with the name *client2\_data*, type:

```
mksp -fb client2_data -sp client_data -size 100m
```

The new storage pool is created with the name *client2\_data*.

---

## IVM mksvcevent command

### Purpose

Creates a new serviceable event. This command is valid only in an Integrated Virtualization Manager environment.

### Syntax

```
mksvcevent -d Description --reporting_mtms ReportingMTMS
```

### Description

The **mksvcevent** command creates a serviceable event with the specified description. This event will show up in the list of serviceable events obtained by the **lssvcevents** command.

## Flags

|                                                |                                                                                                                                                                                                       |
|------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-d</b> <i>Description</i>                   | The description or text of the event.                                                                                                                                                                 |
| <b>-reporting_mtms</b><br><i>ReportingMTMS</i> | The type-model*serial of the reporting system. This should be in the form tttt-mmm*sssssss, where tttt is the machine type, mmm is the model, and sssssss is the serial number of the managed system. |

## Exit Status

This command has a return code of zero on success.

## Security

This command is not accessible by users with the ViewOnly role.

## Examples

1. To create a serviceable event, type:

```
mksvcevent -d This is a test event entry -reporting_mtms 9111-520*XXXXXXX
```

## Related Information

The **lssvcevents** command, and the **chsvcevent** command.

---

## IVM mksyscfg command

### Purpose

Creates a logical partition on the managed system.

### Syntax

To create a logical partition on the managed system

```
mksyscfg -r lpar { -f ConfigurationFile | -i ConfigurationData } [-m ManagedSystem]
```

### Description

The **mksyscfg** command creates a logical partition on the managed system.

### Flags

|                                |                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-r</b> <i>ResourceType</i>  | The type of resources to create:                                                                                                                                                                                                                                                                                                                                                                      |
| <b>-m</b> <i>ManagedSystem</i> | <b>lpar</b> - Logical partition resources<br>The name of the managed system. This attribute is optional because there is only one system to manage. The name might either be the user-defined name for the managed system, or be in the form <i>tttt-mmm*sssssss</i> , where <i>tttt</i> is the machine type, <i>mmm</i> is the model, and <i>sssssss</i> is the serial number of the managed system. |

## **-f** *ConfigurationFile*

The name of the file containing the configuration data needed to change the resources. The configuration data consists of attribute name/value pairs, which are in comma separated value (CSV) format. These attribute name/value pairs form a configuration record. A line feed marks the end of a configuration record. The file must contain one configuration record for each resource to be changed, and each configuration record must be for the same resource type. If the resource type is the managed system or the managed frame, then the file must contain only one configuration record.

The format of a configuration record is as follows:

```
attribute-name=value,attribute-name=value,...<LF>
```

Note that certain attributes accept a comma separated list of values, as follows:

```
"attribute-name=value,value,...",...<LF>
```

When a list of values is specified, the attribute name/value pair must be enclosed in double quotation marks. Depending on the shell being used, nested double quotation marks might need to be preceded by an escape character.

### **Required Attributes for Partition**

#### **desired\_mem**

The assigned megabytes of memory for this partition

#### **lpar\_env**

A required attribute that indicates the type of partition to create. This attribute is used to create an RPA partition, which supports AIX, Linux, or IBM i partition types.

**name** Name of the partition to create.

#### **max\_mem**

The maximum megabytes of memory for this partition

#### **min\_mem**

The minimum megabytes of memory for this partition.

### **Optional Attributes for partitions**

#### **allow\_perf\_collection**

Permission for the partition to retrieve shared processor pool utilization information. Valid values are:

- 0: do not allow authority
- 1: allow authority

#### **alt\_restart\_device\_slot**

The location of the virtual I/O slot that contains the alternate restart device for the IBM i logical partition. If the load source slot is a value other than none, then this attribute is optional. Valid values are:

- Slot Number (virtual I/O)
- none

**alt\_console\_slot**

The location of the physical I/O slot that contains the alternate console device for the IBM i logical partition. The default value is none.

**auto\_start**

Valid values are:

0 - do not automatically start with system power on

1 - automatically start with system power on

**boot\_mode**

Partition power on mode. Valid values are:

norm - normal

dd - diagnostic with default boot list

ds - diagnostic with stored boot list

of - Open Firmware OK prompt

sms - System Management Services

**console\_slot**

The location of the virtual I/O slot that contains the console device for the IBM i logical partition. Valid values are:

- Slot Number (virtual I/O)

- none

**desired\_io\_entitled\_mem**

The amount of I/O entitled memory for a shared memory partition. This is the portion of memory that is reserved for I/O mappings.

- auto (automatically manage)

- *Number of megabytes*

If the value is auto, the entitlement is calculated based on the virtual I/O configuration of the logical partition. If the virtual I/O configuration is changed, the entitlement is updated automatically. If auto is not used, no automatic adjustments are made. The default value is auto.

**desired\_proc\_units**

The assigned number of processing units for this partition

**desired\_procs**

The assigned number of processors for this partition. In shared processing mode, this refers to virtual processors.

**io\_slots**

Comma separated list of I/O slots for the partition. Each item in this list has the format: *drc\_index/slot\_io\_pool\_id/is\_required*

The attribute names are not present in the list, just their values are present. For example, *21010003/none/1*, specifies an I/O slot with a DRC index of 0x21010003 which is not assigned to an I/O pool, and it is a required slot.

Valid values for *is\_required*:

- 0 - no
- 1 - yes

**ipl\_source**

The IPL source for the IBM i logical partition. This attribute is optional. Valid values are:

- a
- b
- c
- d (default)

**lhea\_capabilities**

Comma-separated list of logical Host Ethernet Adapter capabilities, with each capability having one of the following formats: *adapter-ID/capability* or *adapter-ID/5/ieq/nieq/qp/cq/mr* where *ieq* (interruptible event queues), *niesq* (non-interruptible event queues), *qp* (queue pairs), *cq* (completion queues), and *mr* (memory regions) each specify the resource amount in addition to the base minimum. Valid values are:

- 0 - minimum
- 1 - low
- 2 - medium
- 3 - high
- 4 - dedicated
- 5 - custom

**lhea\_logical\_ports**

Comma-separated list of Logical Host Ethernet Adapter (LHEA) logical ports, with each logical port having the following format:

*adapter-ID/port-group/physical-port-ID/  
logical-port-ID/allowed-VLAN-IDs*

All four '/' characters must be present, but optional values may be omitted. Optional values are *allowed-VLAN-IDs*.

**load\_source\_slot**

The location of the virtual I/O slot that contains the load source for the IBM i logical partition. If the `alt_restart_device_slot` has a value other than none, then this attribute is optional. Valid values are:

- Slot Number (virtual I/O)
- none

**lpar\_avail\_priority**

The priority of the partition to maintain its entitled processors. If a processor failure occurs, processing resources will be removed first from the lowest priority partition.

**lpar\_id** Unique integer ID for the partition. If this attribute is not specified, the lowest available partition will be assigned.

**lpar\_proc\_compat\_mode**

The requested compatibility mode. Use `lssyscfg -r sys -F lpar_proc_compat_modes` to retrieve a list of valid values.

**max\_procs**

The maximum number of processors for this partition. In shared processing mode, this refers to virtual processors.

**max\_proc\_units**

The maximum number of processing units for this partition

**max\_virtual\_slots**

Maximum number of virtual I/O adapter slots.

**Note:** The system determines this value.

**mem\_mode**

The partition memory mode.

- ded - dedicated memory
- shared - shared memory

If the memory mode is shared, then the logical partition cannot be assigned any physical I/O slots or host Ethernet adapter resources, the `proc_mode` attribute on the logical partition must be shared, and a memory pool must exist. If the memory mode is omitted, then it is assumed to be a dedicated memory mode.

**Note:** The Virtual I/O Server logical partition only supports the dedicated memory mode.

**mem\_weight**

The shared memory weight of the logical partition that uses shared memory.

This attribute is used for determining the priority of logical partitions in a memory pool for distribution of memory. The weight must be a value from 0 - 255. If a value is not specified, the default is 128.

**min\_procs**

The minimum number of processors for this partition. In shared processing mode, this refers to virtual processors.

**min\_proc\_units**

The minimum number of processing units for this partition



**op\_console\_slot**

The location of the virtual I/O slot that contains the directly-attached operations console device for the IBM i logical partition. The default value is none.

**paging\_device**

The paging space device to use if a memory pool is used. A paging device is a block storage device that has been added to the memory pool and is not designated as a paging device for any other logical partition. This attribute is optional. If omitted, an appropriate paging device is selected automatically. If the paging\_device value is a blank string, no device is assigned.

**proc\_mode**

Valid values are:  
ded- dedicated processor mode  
shared- shared processor mode

**profile\_name**

Name of the profile to create. This attribute is not required, but if specified, must be the same as the **name** attribute.

**sharing\_mode**

Value values are:  
keep\_idle\_procs- never share processors  
share\_idle\_procs- share processors  
only when the partition  
is inactive  
share\_idle\_procs\_always- share processors  
only when the partition is active  
share\_idle\_procs\_active- always share processors  
cap- capped mode  
uncap- uncapped mode

**uncap\_weight**

A weighted average of processing priority when in uncapped sharing mode. The smaller the value, the lower the weight. Valid values are: 0 - 255

**virtual\_eth\_adapters**

Comma-separated list of virtual Ethernet adapters, with each adapter having the following format:

```
slot_number/is_ieee/port_vlan_id/additional_vlan_ids/
is_trunk/is_required
```

All 5 '/' characters must be present, but optional values might be omitted. Optional values are additional\_vlan\_ids, and is\_trunk. Valid values for is\_ieee, is\_trunk, and is\_required:

0 - no  
1 - yes

For example, 4/0/2//0/0 specifies a virtual Ethernet adapter with a virtual slot number of 4, is not IEEE 802.1Q enabled, has a port virtual LAN ID of 2, no additional virtual LAN IDs, it is not a trunk adapter, and is not required.

### **virtual\_fc\_adapters**

Comma-separated list of virtual fibre channel adapters. Each item in this list has the following format:

```
virtual_slot_num/adapter_type/remote_lpar_id/
remote_lpar_name/remote_slot_num/wwpn_list/is_required
```

**Required values:** remote\_lpar\_id, remote\_lpar\_name, adapter\_type, virtual\_slot\_num

**Note:** You can specify either remote\_lpar\_id, remote\_lpar\_name, or use them both, but at least one of the values is required.

Valid values for adapter\_type:

- client
- server

**Note:** If you specify a value for adapter type, the Integrated Virtualization Manager (IVM) requires the adapter type to be a client.

**Optional values:** wwpn\_list, is\_required, remote\_slot\_num

When you add a virtual fibre channel adapter, wwpn\_list can be left blank to allow IVM to automatically assign worldwide port names to the client adapter. If you leave wwpn\_list blank and the virtual slot number specified for this adapter already contains a virtual fibre channel adapter, IVM will use the worldwide port names that are already assigned. New worldwide port names will only be generated if it is a new adapter. If you specify a wwpn\_list value, there must be exactly two values. Each worldwide port name must be a 16-character hexadecimal value. These values are not case sensitive.

Valid values for is\_required:

- 0 - no
- 1 - yes

A value of none or an empty string indicates that no virtual fibre channel adapters should be assigned.

**Note:** If the logical partition being changed is the Virtual I/O Server, you cannot change the current configuration. The IVM handles the client and server adapter as a pair; therefore, IVM handles changes automatically.

### **virtual\_scsi\_adapters**

Comma-separated list of virtual SCSI adapters. Each item in this list has the format:

```
slot_num/adapter_type/remote_lpar_id/remote_lpar_name/
remote_slot_num/is_required
```

The attribute names are not present in the list, just their values are present. If an attribute is optional and is not to be included, then no value is specified for that attribute. For example, `2/client//lpar2/3/0` specifies a virtual client SCSI adapter with a virtual slot number of 2, a server partition name of `lpar2`, a server slot number of 3, and is not required. The server partition ID was omitted.

**Required values:** `slot_num`, `adapter_type`, `remote_lpar_id`, `remote_lpar_name`

**Note:** You can specify `remote_lpar_id`, `remote_lpar_name`, or both, but at least one of the values is required.

**Optional values:** `is_required`, `remote_slot_num`

**Note:** IVM requires that virtual slot number 2 always contain a virtual SCSI adapter, so if you specify adapters in any other slots, the default adapter is still created in slot number 2. If you use the `chsyscfg` command with an empty list for the `virtual_scsi_adapters` attribute, all virtual SCSI adapters are removed, except the default adapter.

Valid values for `adapter_type`:

- `client`: client adapter
- `server`: server adapter, valid for Virtual I/O Server logical partitions only

Valid values for `is_required`:

- 0 - no
- 1 - yes

### **work\_group\_id**

Valid values:

- `none`: do not participate in the workload management group
- `1`: participate in the workload management group

### **-i ConfigurationData**

This option allows you to enter configuration data on the command line, instead of using a file. Data entered on the command line must follow the same format as data in a file, and must be enclosed in double quotation marks. When this option is used, only a single resource can be changed. The `-i` and the `-f` options are mutually exclusive.

## **Exit Status**

This command has a return code of 0 on success.

## **Security**

This command is not accessible by users with the `ViewOnly` role.

## **Examples**

1. To create a partition named `lp3` with 128 megabytes, type:  

```
mksyscfg -r lpar -i "name=lp3,lpar_env=aixlinux,min_mem=128,desired_mem=128,\
max_mem=128"
```
2. To create a partition with 128 megabytes and a dedicated processor, type:

```
mksyscfg -r lpar -i "name=lp4,lpar_env=aixlinux,min_mem=128,desired_mem=128,\
max_mem=128,proc_mode=ded, sharing_mode=share_idle_procs,min_procs=1,\
desired_procs=1,max_procs=2"
```

3. To create a partition with 128 megabytes and 0.2 shared processing units, type:

```
mksyscfg -r lpar -i "name=lp2,lpar_env=aixlinux,min_mem=128,desired_mem=128,\
max_mem=128,proc_mode=shared, sharing_mode=uncap,min_procs=1,\
desired_procs=1,max_procs=2,min_proc_units=0.1,desired_proc_units=0.2,\
max_proc_units=2"
```

4. To create a partition with 128 megabytes and 0.2 shared processing units, and a virtual Ethernet adapter on VLAN 1, type:

```
mksyscfg -r lpar -i "name=lp2,lpar_env=aixlinux,min_mem=128,desired_mem=128,\
max_mem=128,proc_mode=shared, sharing_mode=uncap,min_procs=1,desired_procs=1,\
max_procs=2,min_proc_units=0.1,desired_proc_units=0.2, max_proc_units=2,\
virtual_eth_adapters=4/0/1//0/0"
```

5. To create a partition with 128 megabytes and 0.2 shared processing units, a virtual Ethernet adapter on VLAN 1, and HEA ports 3 and 4, type:

```
mksyscfg -r lpar -i "name=lp2,lpar_env=aixlinux,min_mem=128,desired_mem=128,\
max_mem=128,proc_mode=shared, sharing_mode=uncap,min_procs=1,desired_procs=1,\
max_procs=2,min_proc_units=0.1,desired_proc_units=0.2, max_proc_units=2,\
virtual_eth_adapters=4/0/1//0/0, \ \
"lhea_logical_ports=23000000/1/0/3/all,23000000/1/1/4/all", \
lhea_capabilities=23000000/0/////"
```

6. To create a logical partition with 1 gigabyte, 2 virtual processors, and 2 virtual Ethernet adapters, type:

```
mksyscfg -r lpar -i 'name=lp2,lpar_env=aixlinux,min_mem=256,desired_mem=1024,\
max_mem=2048,proc_mode=shared,sharing_mode=uncap,min_procs=1,desired_procs=2,\
max_procs=2,min_proc_units=0.1,desired_proc_units=0.2,max_proc_units=2,\
"virtual_eth_adapters=4/0/1//0/0,""5/1/2/212,313/0/0""'
```

## Related Information

The `lssyscfg` command, the `chsyscfg` command, and the `rmsyscfg` command.

---

## IVM mksysplan command

### Purpose

Creates an system plan file that represents the information known about a managed system's hardware, partitions, profiles, and partition provisioning information. The `mksysplan` command will also have information about Virtual I/O Devices.

### Syntax

```
mksysplan -f <file name> -m <managed system> [--check] [-d "<description>"] [--noinvscout] [--help]
```

### Description

The `mksysplan` command creates a system plan that represents the information known about a managed system's hardware, partitions, profiles, and partition provisioning.

### Flags

|                                        |                                                                                                                                                                                                                                                                                                                                                        |
|----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-f</b> <i>system plan file name</i> | Specifies the system plan file name that will contain the system plan created by the command. If it does not exist, the file will be created in the system plan file directory on the Integrated Virtualization Manager partition. If the file exists the file contents will be overwritten. The file name must end in a <code>.sysplan</code> suffix. |
|----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|                                 |                                                                                                                                                                                                                                                                 |
|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-m</b> <i>managed system</i> | Specifies the managed system's name where the plan should be created. Since the Integrated Virtualization Manager can work with only one managed system, <b>-m</b> flag should specify the default managed system only.                                         |
| <b>--check</b>                  | Specifies that the command should check the system plan directory and fail with an error if the file with the name specified by <b>-f</b> already exists.                                                                                                       |
| <b>-d</b> <i>description</i>    | Specifies a description that will be added to the generated file.                                                                                                                                                                                               |
| <b>--noinvscout</b>             | Attempts the hardware discovery for unallocated or inactive partition hardware. The <b>--noinvscout</b> flag does not perform additional inventory probes for active partitions.<br><b>Note:</b> The <b>--noprobe</b> flag overrides the action of this option. |
| <b>--help</b>                   | Displays the help text for this command and then exits.                                                                                                                                                                                                         |

## Exit Status

See "Exit status for Virtual I/O Server commands" on page 1.

## Examples

1. To make a system plan file *sample.sysplan* that represents the configuration of the Integrated Virtualization Manager system, type the following command:

```
mksysplan -f sample.sysplan
```

## Related Information

The **deploysysplan** command, **lssysplan** command, and the **rmsysplan** command.

---

## mktcpip command

### Purpose

Sets the required values for starting TCP/IP on a host.

### Syntax

```
mktcpip -hostname HostName -inetaddr Address -interface Interface [-start] [-netmask SubnetMask] [-cabletype CableType] [-gateway Gateway] [-nsrvaddr NameServerAddress -nsrvdomain Domain]
```

To run stateless autoconfiguration of link-local addresses:

```
mktcpip [-interface Interface] -auto
```

To add a static IPv6 address:

```
mktcpip -interface Interface -inetaddr IPv6 address [-plen PrefixLen] [-gateway gateway]
```

### Description

The **mktcpip** command sets the minimal values required for using TCP/IP on a host machine. The basic functions of the **mktcpip** command include the following:

- Setting the host name
- Setting the IP address of the interface
- Setting the domain name and IP address of the nameserver, if applicable
- Setting the subnetwork mask, if applicable
- Starting the specified TCP/IP daemons

## Flags

|                                              |                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-auto</b>                                 | Enables automatic stateless configuration of link-local addresses. Link-local addresses can coexist with other preexisting IPv6 addresses. Therefore, if an address is assigned using the <b>-auto</b> flag, it will not replace existing IPv6 addresses.                                                                                                                                                     |
| <b>-cabletype</b> <i>CableType</i>           | Specifies the cable size for Standard Ethernet or IEEE 802.3 Ethernet networks. Valid values for the <i>CableType</i> variable are <i>dix</i> for thick cable, <i>bnc</i> for thin cable, or <i>N/A</i> for not applicable. The <b>-cabletype</b> <i>CableType</i> flag should be used only for Standard Ethernet ( <i>en</i> ) and IEEE 802.3 Ethernet ( <i>et</i> ) interfaces. The default is <i>N/A</i> . |
| <b>-gateway</b> <i>Gateway</i>               | Sets the gateway address for a static route. Specify the address in dotted decimal notation.                                                                                                                                                                                                                                                                                                                  |
| <b>-hostname</b> <i>Hostname</i>             | Sets the name of the host. If using a domain naming system, the domain and any subdomains must be specified. The following is the standard format for setting the host name:<br><br>hostname<br><br>The following is the standard format for setting the host name in a domain naming system:<br><br>hostname.subdomain.subdomain.rootdomain                                                                  |
| <b>-inetaddr</b> <i>Address</i>              | Sets the Internet address of the host. Specify the address in dotted decimal notation. Each network interface on the host should have a unique Internet address. The following is the standard format for setting the Internet address:<br><br>127.10.31.2                                                                                                                                                    |
| <b>-interface</b> <i>Interface</i>           | Specifies a particular network interface, for example:<br><br>en1                                                                                                                                                                                                                                                                                                                                             |
| <b>-netmask</b> <i>SubnetMask</i>            | Specifies the mask that the gateway should use in determining the appropriate subnetwork for routing. The subnet mask is a set of 4 bytes, as in the Internet address. The subnet mask consists of high bits (1s) corresponding to the bit positions of the network and subnetwork address, and low bits (0s) corresponding to the bit positions of the host address.                                         |
| <b>-nsrvaddr</b><br><i>NameserverAddress</i> | Specifies the Internet address of the name server that the host uses for name resolution, if applicable. The address should be entered in dotted decimal notation, as follows:<br><br>127.1.0.1                                                                                                                                                                                                               |
| <b>-nsrvdomain</b> <i>Domain</i>             | Specifies the domain name of the name server that the host should use for name resolution, if any. The domain name should be in the following format:<br><br>subdomain.subdomain.rootdomain                                                                                                                                                                                                                   |
| <b>-plen</b> <i>prefixLen</i>                | Specifies the prefix length of the IPv6 interface.                                                                                                                                                                                                                                                                                                                                                            |
| <b>-start</b>                                | Starts the TCP/IP daemons.                                                                                                                                                                                                                                                                                                                                                                                    |

## Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

## Examples

1. To set the required values for starting TCP/IP, type:  

```
mktcpip -hostname fred.austin.century.com -inetaddr 192.9.200.4 -interface en0 \
-nsrvaddr 192.9.200.1 -nsrvdomain austin.century.com -start
```
2. To set the required values for starting TCP/IP using an IPv6 address, type:  

```
mktcpip -interface en0 -auto
```

**Note:** If an IPv6 address already exists on the specified interface, this command format will not replace it. You can use the **-auto** flag to allow multiple IPv6 addresses to coexist on the interface.

3. To set the required values for starting TCP/IP, using an IPv6 address to replace any pre-existing IPv6 addresses, type:  

```
mktcpip -interface en0 -hostname host -inetaddr ipv6_address
```

**Note:** This method replaces any IPv6 addresses that were assigned to the interface.

## Related Information

The **hostname** command, the **startnetsvc** command, the **stopnetsvc** command, the **cfglnagg** command, the **netstat** command, the **entstat** command, the **cfgnamesrv** command, the **hostmap** command, the **traceroute** command, the **ping** command, the **optimizenet** command.

---

## mkuser command

### Purpose

Creates a new user account.

### Syntax

```
mkuser [-ldap] [-de | -sr] [-attr Attribute=Value [Attribute=Value...]] Name
```

### Description

The **mkuser** command creates a new user account. Upon completion of creating a the new account you are prompted to set the new account password. User accounts are created with the **-attr pgrp=view** are designated as read only. These users do not have the authority to change the system configuration and do not have write permission to their home directories.

### Flags

|                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-attr</b> <i>Attribute=Value</i> | Identifies the attribute to set, as well as the new value for the attribute. The <i>Attribute=Value</i> parameter can use one attribute value pair or multiple attribute value pairs for one <b>-attr</b> flag.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>-de</b>                          | For a complete list of supported attributes, see “chuser command” on page 75.<br>Creates a Development Engineer (DE) user account. This type of account allows IBM developers to log into the Virtual I/O Server and debug problems.                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>-ldap</b>                        | Identifies the user as an LDAP user account. LDAP user accounts are authenticated through the LDAP load module.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>-sr</b>                          | Creates a service representative (SR) user account. This type of account enables a service representative to run commands required to service the system without being logged in as root. This includes the following command types: <ul style="list-style-type: none"><li>• Run diagnostics, including service aids (for example, hot plug tasks, certify, format, and so forth).</li><li>• Run all commands that can be run by a group system.</li><li>• Configure and unconfigure devices that are not busy.</li><li>• Use the service aid to update system microcode.</li><li>• Perform the shut down and reboot operations.</li></ul> The recommended SR login user name is <b>qserv</b> . |

### Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

### Security

This command can only be run by the prime administrator (padmin) user.

## Examples

1. To create the **davis** user account with default values, type the following command:  
`mkuser davis`
2. To create the **davis** user account and set the **maxage** attribute to a value of 52, type the following command:  
`mkuser -attr maxage=52 davis`
3. To create a user with read only authority, type the following command:  
`mkuser -attr pgrp=view View1`

## Related Information

The `chuser` command, the `lsuser` command, the `rmuser` command, and the `passwd` command.

---

## mkvdev command

### Purpose

Adds a virtual device to the system.

### Syntax

To create a virtual target device:

```
mkvdev [-f] { -vdev TargetDevice | -dplc TDPhysicalLocatonCode } { -vadapter VirtualServerAdapter | -aplc VSAPhysicalLocationCode } [-dev DeviceName]
```

To create a virtual target device for a file backed virtual optical device:

```
mkvdev -fbo { -vadapter VirtualServerAdapter | -aplc VSAPhysicalLocationCode } [-dev DeviceName]
```

To create a shared Ethernet Adapter:

```
mkvdev -sea TargetDevice -vadapter VirtualEthernetAdapter... -default DefaultVirtualEthernetAdapter -defaultid SEADefaultPVID [-attr Attribute=Value [Attribute=Value...]]
```

To create an Link Aggregation adapter:

```
mkvdev -lnagg TargetAdapter... [-attr Attribute=Value [Attribute=Value...]]
```

To create a VLAN Ethernet adapter:

```
mkvdev -vlan TargetAdapter -tagid TagID
```

### Description

The **mkvdev** command creates a virtual device. The name of the virtual device is automatically generated and assigned unless the **-dev** *DeviceName* flag is specified, in which case *DeviceName* becomes the device name.

If the **-lnagg** flag is specified, a Link Aggregation or IEEE 802.3 Link Aggregation (automatic Link Aggregation) device is created. To create an IEEE 802.3 Link Aggregation set the mode attribute to 8023ad. If the **-sea** flag is specified, a Shared Ethernet Adapter is created. The *TargetDevice* may be a Link Aggregation adapter (note, however, that the *VirtualEthernetAdapter* may not be Link Aggregation adapters). The default virtual Ethernet adapter, *DefaultVirtualEthernetAapter*, must also be included as one of the virtual Ethernet adapters, *VirtualEthernetAdapter*.



The **-fbo** flag is used to create a virtual target device which maps the *VirtualServerAdapter* to a file backed virtual optical device. Virtual optical devices cannot be used until virtual media is loaded into the device. See the **loadopt** command for details.

The **-vlan** flag is used to create a VLAN device and the **-vdev** flag creates a virtual target device which maps the *VirtualServerAdapter* to the *TargetDevice*.

If the backing device that is specified by the **-vdev** or **-dplc** flags is already in use, an error is returned unless the **-f** flag is also specified. Also, the backing device that is specified cannot be a physical or logical volume that is assigned to a shared memory pool (to be used as a paging space device by a shared memory partition).

The **mkvdev** command also configures virtual optical and tape devices, where the **-vdev** or **-dplc** flag specifies the physical optical or tape device and the **-vadapter** or **-aplc** flag specifies the virtual SCSI adapter. If the specified optical or tape device is already assigned to a virtual SCSI adapter, an error is returned unless the **-f** flag is also specified. If the **-f** flag is specified, the optical or tape device will be removed from the virtual SCSI adapter it is currently assigned to before reassigning it to the new virtual SCSI adapter.

When an additional disk drive that has a lower *max\_transfer\_size* than the current setting is added, and it is set as a virtual target device (processing **mkvdev**), the client cannot recognize this device until the VIOS is rebooted and the *max\_transfer\_size* is reestablished to the new setting. Because the *max\_transfer\_size* cannot be dynamically changed, the **mkvdev** command checks the current *max\_transfer\_size* of the device being added. If it is lower, it posts a message stating that the VIOS must be rebooted before the client can see this device.

**Attention:** To protect the Configuration Database, the **mkvdev** command is not interruptible. Stopping this command before execution is complete could result in a corrupted database.

## Flags

|                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-aplc</b> <i>VSAPhysicalLocationCode</i>          | Specifies the virtual SCSI adapter using the physical location code                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>-attr</b> <i>Attribute=Value</i>                  | Specifies the device attribute value pairs to be used instead of the defaults. The <i>Attribute=Value</i> variable can be used to specify one attribute value pair or multiple attribute value pairs for one <b>-attr</b> flag. If you use an <b>-attr</b> flag with multiple attribute value pairs, the list of pairs must be enclosed in quotation marks with a blank space between the pairs. For example, entering <b>-attr Attribute=Value</b> lists one attribute value pair per flag, while entering <b>-attr 'Attribute1=Value1 Attribute2=Value2'</b> lists more than one attribute value pair. |
| <b>-default</b> <i>DefaultVirtualEthernetAdapter</i> | Default virtual adapter to use for non-VLAN-tagged packets. This flag maps to the SEA device attribute <i>pvid_adapter</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>-defaultid</b> <i>SEADefaultPVID</i>              | The <i>SEADefaultPVID</i> is the VID used for untagged frames. All untagged packets are assigned the <i>SEADefaultPVID</i> value. When a tagged frame is received by a port, the tag is used. Otherwise if the frame is untagged, the value contained in the PVID is considered as a tag. This flag maps to the SEA device attribute <i>pvid</i> .                                                                                                                                                                                                                                                       |
| <b>-dev</b> <i>DeviceName</i>                        | By using the <b>-dev</b> flag, you can specify the name you want the device to be known by. If you do not use the <b>-dev</b> flag, a name will be automatically generated and assigned. Not all devices support user-supplied names.                                                                                                                                                                                                                                                                                                                                                                    |
| <b>-dplc</b> <i>TDPhysicalLocatonCode</i>            | Specifies the physical device using the physical location code                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>-f</b>                                            | Force the physical volume specified to be used as a backing device even if its already associated with a virtual SCSI adapter. If the specified backing device is an optical device, <b>-f</b> forces the optical device will be removed from the virtual SCSI adapter it is currently assigned to before reassigning it to the new virtual SCSI adapter.                                                                                                                                                                                                                                                |
| <b>-fbo</b>                                          | Create a virtual optical device.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

|                                                                               |                                                                                                                                                                                                                                                                                                                                |
|-------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-lnagg</b> <i>TargetAdapter...</i>                                         | Creates a Link Aggregation device.                                                                                                                                                                                                                                                                                             |
| <b>-sea</b> <i>TargetDevice</i>                                               | Creates a Shared Ethernet Adapter which maps <i>VirtualEthernetAdapter</i> to the adapter <i>TargetDevice</i> . <i>TargetDevice</i> can be a physical adapter or a Link Aggregation adapter.                                                                                                                                   |
| <b>-tagid</b> <i>TagID</i>                                                    | Specifies the VLAN tag ID.                                                                                                                                                                                                                                                                                                     |
| <b>-vadapter</b> <i>VirtualEthernetAdapter</i> or <i>VirtualServerAdapter</i> | Specifies the virtual server adapter or virtual Ethernet adapter the new device will be mapped to. If multiple virtual Ethernet adapters are specified, separate the adapter names with commas and no spaces.                                                                                                                  |
| <b>-vdev</b> <i>TargetDevice</i>                                              | Creates a virtual device mapped to the physical/logical device <i>TargetDevice</i> and the virtual server adapter <i>VirtualServerAdapter</i> . The <i>TargetDevice</i> can be either a physical volume, logical volume, tape, or optical device. Physical volumes assigned to volume groups cannot be used as target devices. |
| <b>-vlan</b> <i>TargetAdapter</i>                                             | Creates a Virtual Local Area Network device.                                                                                                                                                                                                                                                                                   |

## Exit Status

|    |                                                                       |
|----|-----------------------------------------------------------------------|
| 13 | Specified physical or logical volume is not a valid.                  |
| 21 | Device is already in use. Use the <b>-f</b> flag to force assignment. |
| 22 | Logical volumes cannot be assigned as backing devices more than once. |

## Examples

1. To create a virtual target device that maps the logical volume **lv20** as a virtual disk for a client partition hosted by the **vhost0** virtual server adapter, type the following command:

```
mkvdev -vdev lv20 -vadapter vhost0
```

The system displays a message similar to the following:

```
vtscsi0 available
```

2. To create a virtual target device that maps the physical volume **hdisk6** as a virtual disk for a client partition served by the **vhost2** virtual server adapter, type the following command:

```
mkvdev -vdev hdisk6 -vadapter vhost2
```

The system displays a message similar to the following:

```
vtscsi1 available
```

3. To create a virtual target device that maps the physical tape device **rmt0** as a virtual tape device for a client partition served by the **vhost2** virtual server adapter, type the following command:

```
mkvdev -vdev rmt0 -vadapter vhost2
```

The system displays a message similar to the following:

```
vttape0 available
```

4. To create a Shared Ethernet Adapter that maps the physical Ethernet adapter **ent4** as a virtual Ethernet adapter for the client partitions served by the virtual Ethernet adapters **ent6**, **ent7**, and **ent9**, using **ent6** as the default adapter and **8** as the default ID, type the following command:

```
mkvdev -sea ent4 -vadapter ent6,ent7,ent9 -default ent6 -defaultid 8
```

The system displays a message similar to the following:

```
ent10 available
```

5. To create an automatic Link Aggregation with primary adapters **ent4** and **ent5** and backup adapter **ent6**, type the following command:

```
mkvdev -lnagg ent4,ent5 -attr backup_adapter=ent6 mode=8023ad
```

The system displays a message similar to the following:

```
ent10 available
```

## Related Information

The `cfgdev` command, the `chdev` command, the `chpath` command, the `lsdev` command, the `lsmmap` command, and the `rmdev` command.

---

## mkvg command

### Purpose

Creates a volume group.

### Syntax

```
mkvg [-f][-vg VolumeGroup] PhysicalVolume ...
```

### Description

The **mkvg** command creates a new volume group using the physical volumes represented by the *PhysicalVolume* parameter. After creating the volume group, the **mkvg** command automatically activates the new volume group using the **activatevg** command.

#### Note:

1. The physical volume is checked to verify that it is not already in another volume group. If the system believes the physical volume belongs to a volume group that is active, it exits. But if the system detects a description area from a volume group that is not active, it prompts the user for confirmation in continuing with the command. The previous contents of the physical volume are lost, so the user must be cautious when using the override function.
2. This command will fail to add a disk to the volume group if the disk indicates that it is managed by a third party volume manager.
3. The physical volume that you specify cannot be assigned to a shared memory pool (to be used as a paging space device by a shared memory partition).

### Flags

|                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-f</b>                     | Forces the volume group to be created on the specified physical volume unless the physical volume is part of another volume group in the Device Configuration Database or a volume group that is active.                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>-vg</b> <i>VolumeGroup</i> | Specifies the volume group name rather than having the name generated automatically. Volume group names must be unique system wide and can range from 1 to 15 characters. The name cannot begin with a prefix already defined in the PdDv class in the Device Configuration database for other devices. The volume group name created is sent to standard output.<br><br>The volume group name can only contain the following characters: "A" through "Z," "a" through "z," "0" through "9," or "_" (the underscore), "-" (the minus sign), or "." (the period). All other characters are considered invalid. |

### Exit Status

See "Exit status for Virtual I/O Server commands" on page 1.

### Examples

1. To create a volume group that contains physical disks **hdisk3**, **hdisk5**, and **hdisk6**, type:  

```
mkvg hdisk3 hdisk5 hdisk6
```

The volume group is created with an automatically generated name, which is displayed.

2. To create the volume group **newvg** with one physical partition, type:

```
mkvg -vg newvg hdisk1
```

## Related Information

The **lsvg** command, the **chvg** command, the **extendvg** command, the **reducevg** command, the **mirrorios** command, the **unmirrorios** command, the **activatevg** command, the **deactivatevg** command, the **importvg** command, the **exportvg** command, and the **syncvg** command.

---

## mkvopt command

### Purpose

Create a virtual optical media disk in the Virtual Media Repository

### Syntax

```
mkvopt -name FileName {-size Size | -dev OptDevice | -file SourceFile} [-ro]
```

### Description

The **mkvopt** command creates a new virtual optical disk in the Virtual Media Repository. If the **-size** flag is specified, the new optical disk is initialized to all zeros of the given *Size*. If the **-dev** flag is specified, the given device, *OptDevice*, must be an optical device with media loaded. The contents of the loaded media within the device, will be used to create the optical media. If the **-file** flag is specified, *SourceFile* is copied into the repository. By default, the virtual optical disk is created as DVD-RAM media. If the **-ro** flag is specified, the disk will be create as DVD-ROM media.

When specifying *Size* the following conventions must be used:

| Size   | Minimum file size |
|--------|-------------------|
| ###M/m | ###MB             |
| ###G/g | ###GB             |

### Flags

|                                |                                                                          |
|--------------------------------|--------------------------------------------------------------------------|
| <b>-dev</b> <i>OptDevice</i>   | Specifies a physical optical device with media loaded.                   |
| <b>-file</b> <i>SourceFile</i> | Specifies the name of an existing file to by copied into the repository. |
| <b>-name</b> <i>FileName</i>   | Specifies the name of the new virtual optical media file.                |
| <b>-ro</b>                     | Create the new disk as read-only (DVD-ROM).                              |
| <b>-size</b> <i>Size</i>       | Specifies how big to make the new virtual optical media.                 |

### Examples

1. To create a read-only virtual optical disk with the name *pressData* from existing optical media in device *cd0*, type the following command:  

```
mkvopt -name pressData -dev cd0 -ro
```
2. To create a new virtual optical disk with the name *blankDVD* with 1 Gigabyte storage capacity, type the following command:  

```
mkvopt -name blankDVD -size 1g
```

---

## mkvt command

### Purpose

Create a virtual terminal connection to a partition.

### Syntax

```
mkvt { -id lparID }
```

### Description

The **mkvt** command opens a virtual terminal connection to the target partition. You can terminate the virtual terminal connection in one of the following ways:

- The virtual terminal contains an escape sequence that allows you to break out of the command. The escape sequence is `<cr>~.`, or more explicitly: the Enter key, the tilde (~), and the period (.).
- You can use the **rmvt** command to force the session to be closed.

A partition can only have one open virtual terminal session.

This command requires additional HMC configuration if used in an HMC environment.

### Flags

**-id** *lparID*                      The ID of the partition for which to open the virtual terminal session.

### Exit Status

|    |                                       |
|----|---------------------------------------|
| 27 | Unexpected error                      |
| 28 | Virtual terminal is already connected |
| 29 | Virtual terminal device is not found  |
| 30 | Permission denied                     |
| 31 | Specified device does not exist       |

### Security

This command is not accessible by users with the ViewOnly role.

### Examples

1. Create a virtual terminal connection to the partition with ID 3:

```
mkvt -id 3
```

### Related Information

The **rmvt** command.

---

## more command

### Purpose

Displays file contents one screen at a time.

## Syntax

```
more [-c] [-d] [-e] [-H] [-i] [-l] [-N] [-s] [-u] [-v] [-z] [-n Number] [-p Subcommand] [-t Tagstring] [-W Option] [-x Tabs] [File ...]
```

## Description

The **more** command reads files and displays the text one screen at a time. The command pauses after each screen and prints the word **More** at the bottom of the screen. If you then press a carriage return, the **more** command displays an additional line. If you press the space bar, the **more** command displays another full screen of text.

**Note:** On some terminal models, the **more** command clears the screen, instead of scrolling.

Instead of naming files to read, you can either redirect or pipe standard output, such as a long directory listing, to the **more** command. The command adds a % (percent sign) to its prompt when reading from a file rather than a pipe. This provides the percentage of the file (in characters, not lines) that the **more** command has read.

The **more** command sets the terminal to NOECHO mode so the output can be continuous. With the exception of the / and ! subcommands, commands that are typed do not normally show up on the terminal. If the standard output is not a terminal, the **more** command will act just like the **cat** command, except that a header will be printed before each file in a series.

## Environment Variables

Environment variables affect the way the **more** command works. You can set some environment characteristics in the `/etc/environment` file and system profile files, such as the `.ksh`, `.csh`, and `.profile` files.

The **more** command uses the **TERM** variable to determine terminal characteristics. If this variable is NULL or not set, the command uses the default terminal type. The `/usr/share/lib/terminfo` directory contains definitions for terminal characteristics.

By default, the **more** command window size is 2 lines less than what the system terminal is capable of. The command sets the default window size based on the **LINES** variable. Also, you can easily adjust the window size for each run of the command by adding the **-n** flag.

Use the **MORE** variable to customize the **more** command with your preferred configuration each time the system starts. This variable accepts **more** command flags.

## Flags

|           |                                                                                                                                                                                                                                                                                                             |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-c</b> | Prevents the screen from scrolling, which makes text easier to read as the <b>more</b> command writes to the screen. The system ignores the <b>-c</b> flag if the terminal cannot clear to the end of a line.                                                                                               |
| <b>-d</b> | Prints a message, appended to the <b>More</b> prompt at the bottom of the screen, about which keys continue, quit, and provide help for the <b>more</b> command. Displays error messages rather than ringing the terminal bell if an unrecognized command is used. This is helpful for inexperienced users. |
| <b>-e</b> | Exits automatically after displaying the last line of the last file.                                                                                                                                                                                                                                        |
| <b>-H</b> | Disables the search pattern highlighting feature by default.                                                                                                                                                                                                                                                |
| <b>-i</b> | Searches for patterns without considering uppercase and lowercase.                                                                                                                                                                                                                                          |
| <b>-l</b> | Pauses after detecting a page break in the input. If the <b>-l</b> flag is not used, the <b>more</b> command pauses to accept commands after any line containing a ^L (CTRL-L) character. Also, if a file begins with a FORMFEED, the screen is cleared before the file is printed.                         |

- N** Suppresses line numbering. The default display, with line numbers, can slow the **more** command's performance on very large input files. The line numbering feature displays the line number in the = subcommand and passes the line number to the editor (if it is the **vi** editor).
- n Number** Configures the **more** command to display the specified number of lines in the window. Without the **-n** flag, the **more** command defaults to two lines less than what the terminal is capable of. For example, on a 24-line terminal, the default is 22 lines. The **-n** option overrides any values obtained from the environment.
- p Subcommand** Starts the **more** command and specified subcommand for each *File* operand. For example, **more -p 50j text1 text2** displays the **text1** file at the fiftieth line; then does the same for the **text2** file when you finish the first. See "Subcommands" for descriptions of **more** subcommands.
- If the command is a positioning command, such as a line number or a regular expression search, set the current position to represent the final results of the command, without writing any intermediate lines of the file. For example, the two commands:
- ```
more -p 1000j filename
more -p 1000G filename
```
- are functionally the same and will start the display with the current position at line 1000, passing the lines that **j** would write and would scroll off the screen if it had been issued during the file examination.
- If the positioning command is unsuccessful, the first line in the file will be the current position.
- s** Reduces multiple blank lines in the output to only one blank line.
- t Tagstring** Displays the portion of the file that contains the specified tag.
- u** Prevents the **more** command from treating a backspace character as a printable control character (displayed as a ^H (CTRL-H)), suppressing backspacing, underlining, or creating reverse video text for underlined information in a source file. The **-u** flag also forces the **more** command to recognize a carriage-return character, if it exists, at the end of a line.
- v** Suppresses the graphical translation of nonprinting characters. Without the **-v** flag, the **more** command graphically interprets all non-ASCII and most control characters, except Tab, Backspace, and Return. For example, if you do not use the **-v** flag, the **more** command displays the non-ASCII characters Ctrl-x as ^X and x as M-x.
- W Option** Provides the specified *Option* to the **more** command as an extension:
- notite** Prevents the **more** command from sending the terminal initialization string (either the **ti termcap** or the **smcup terminfo** capability) before displaying the file. This option also prevents the **more** command from sending the terminal de-initialization string (either the **te termcap** or the **rmcup terminfo** capability) before exiting.
- tite** Causes the **more** command to send the initialization and de-initialization strings. This is the default.
- These options control whether the **more** command sends the initialization strings described, which, for certain terminals (such as some xterms), cause the **more** command to switch to an alternative screen. The effect of switching screens is to erase the display of the file you were viewing.
- x Tabs** Sets tab stops at the specified *Tabs* position. The default tab setting is 8 columns.
- z** Graphically displays the Tab, Backspace, and Return control characters. With the **-z** flag, the **more** command translates the Backspace character as ^H, Return as ^M, and Tab as ^I.

Subcommands

The **more** command accepts subcommands when the command pauses and as parameters for the **-p** flag. Many subcommands take an optional integer, symbolized here by *K*, which you must enter before the

subcommand, with no space between. The **more** command, in the paused state, processes subcommands immediately and does not require you to press the Enter key.

The **more** command uses the following subcommands:

h	Displays a help screen that describes the more subcommands.
v	Starts the vi editor, editing the current file in the current line.
r or ^L	Refreshes the display.
R	Refreshes the display and removes buffered input.
[K](Spacebar)	Moves forward <i>K</i> lines when you press the spacebar. If you do not give a value for <i>K</i> , pressing the spacebar displays the next full screen by default. This spacebar subcommand is the same as [K]f or [K]^F or [K]z .
[K]f or [K]^F or [K]z	Moves forward <i>K</i> lines, or one full screen if you do not give a value for <i>K</i> .
[K]b or [K]^B	Moves backward <i>K</i> lines, or one full screen if you do not give a value for <i>K</i> .
[K]d or [K]^D	Moves forward <i>K</i> lines, or half a screen if you do not give a value for <i>K</i> . If you give a value for <i>K</i> , the more command sets the d and u scroll size to <i>K</i> lines for the session.
[K]u or [K]^U	Moves backward <i>K</i> lines, or half a screen if you do not give a value for <i>K</i> . If you give a value for <i>K</i> , the more command sets the d and u scroll size to <i>K</i> lines for the session.
[K]j or [K](Enter) or [K]^E	Moves forward <i>K</i> lines, or one line if you do not give a value for <i>K</i> .
[K]k or [K]^Y	Moves backward <i>K</i> lines, or one line if you do not give a value for <i>K</i> .
[K]g	Moves to the beginning of the file, unless you give a line number for <i>K</i> . The default for <i>K</i> is line number 1.
[K]G	Moves to the last line in the file, unless you give a line number for <i>K</i> . The default for <i>K</i> is the last line in the file.
[K]p or [K]%	Moves to the point in the file that is <i>K</i> percent of the total file. The default for <i>K</i> is one percent, or the first line in the file.
ma-z	Marks the current position in the file with the specified letter.
'a-z	(Single quotation mark) Moves to the position marked with the specified letter.
''	(Two single quotation marks) Returns to the position from which the last large movement (moving more than a page) command was run. If no such movements have been made, returns to the beginning of the file.
[K]/pattern	(Slash) Searches forward, from the current position, for the specified occurrence of the specified pattern of characters. The default value for <i>K</i> is the first occurrence.
[K]!pattern	(Slash, exclamation mark) Searches forward, from the current position, for the specified occurrence of a line that does not contain the specified pattern of characters. The default value for <i>K</i> is the first occurrence.
[K]?pattern	(Question mark) Searches backward, from the current position, for the specified occurrence of the specified pattern of characters. The default value for <i>K</i> is the first occurrence.
[K]?!pattern	(Question mark, exclamation mark) Searches backward, from the current position, for the specified occurrence of a line that does not contain the specified pattern of characters. The default value for <i>K</i> is the first occurrence.
[K]n	Repeats the last search, specifying an occurrence of the pattern (or an occurrence <i>not</i> containing the pattern if the search subcommand included !). The default value for <i>K</i> is the first occurrence.

:a	Lists the file or files you named in the more command line.
:f or ^G or =	Displays information about the current file: <ul style="list-style-type: none"> • file name • order of the file in the list of files • current line number • current position in the file, given as a percentage • current byte number and total bytes to display.
:e[File] or E[File]	Examines the specified file, provided you named it in the more command line.
[K]:n or [K]N	Examines either the next file (if you do not give a value for <i>K</i>) or the file <i>K</i> number of positions forward in the list of files you named in the more command line.
[K]:p or [K]P	Examines either the previous file (if you do not give a value for <i>K</i>) or the file <i>K</i> number of positions backward in the list of files you named in the more command line.
:t <i>Tagstring</i>	Displays the portion of the file that contains the specified tag. This subcommand works only on files containing tags created with the ctags command. The :t subcommand is the interactive version of the -t flag.
:q or q or Q	Exits the more command.
!:command or !command	Starts the specified command in a new shell.
H	Toggles the search pattern highlighting feature on or off.

Exit Status

This command returns the following exit values:

0	Successful completion.
>0	An error occurred.

Examples

1. To view a file named `myfile`, enter:

```
more myfile
```
2. To view output from the **ls** command, enter:

```
ls -l | more
```
3. To view each file starting at its last screen, enter:

```
more -p G file1 file2
```
4. To view each file with the 100th line at the current position, enter:

```
more -p 100 file1 file2
```

Typically, the current position in a **more** command display is the third line on the screen. In this example, the first line on the screen is the 98th line in the file.

5. To view each file starting with the first line that contains the `foo` string, enter:

```
more -p /foo file1 file2
```

The **more** command displays the line in the current position, the third line on the screen.

Files

<code>/usr/share/lib/terminfo</code>	Indicates the terminal information database.
--------------------------------------	--

Related Information

The `cat` command.

motd command

Purpose

Displays or modifies the partition's message of the day file.

Syntax

`motd`

`motd { -append | -overwrite } { -file Filename | "Message of the day string" }`

Description

The `motd` command writes or appends to the partitions message of the day file. The new message can be specified on the command line or in a file with the `-file` flag. If no flags are specified the current message of the day is displayed.

Flags

<code>-append</code>	Appends the specified message to the current message of the day
<code>-file <i>FileName</i></code>	Replaces the current message of the day with the contents of <i>FileName</i>
<code>-overwrite</code>	Replaces the current message of the day with the specified message

mount command

Purpose

Makes a file system available for use.

Syntax

`mount [[Node:Directory] Directory]`

`mount -cd DeviceDirectory`

Description

The `mount` command instructs the operating system to make a file system available for use at a specified location (the mount point). The `mount` command mounts a file system expressed as a directory using the `Node:Directory` parameter on the directory specified by the `Directory` parameter. After the `mount` command has finished, the directory specified becomes the root directory of the newly mounted file system.

If you enter the mount command without flags, the command displays the following information for the mounted file systems:

- the node (if the mount is remote)
- the object mounted
- the mount point
- the virtual-file-system type
- the time mounted

- any mount options

The `/mnt` directory can be used as a local mount point, or you can create a directory using the `mkdir` command. Any directories created with the `mkdir` command must be a sub-directory of your home directory.

Flags

`-cd` Specifies the `cd` device name on which to mount.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

1. To list the mounted file systems, type:

```
mount
```

This command produces output similar to the following:

node	mounted	mounted	vfs	date	options	over
----	-----	-----	---	-----	-----	-----
	/dev/hd0	/	jfs	Dec 17 08:04	rw, log	=/dev/hd8
	/dev/hd3	/tmp	jfs	Dec 17 08:04	rw, log	=/dev/hd8
	/dev/hd1	/home	jfs	Dec 17 08:06	rw, log	=/dev/hd8
	/dev/hd2	/usr	jfs	Dec 17 08:06	rw, log	=/dev/hd8
sue	/home/local/src	/usr/code	nfs	Dec 17 08:06	ro, log	=/dev/hd8

For each file system, the `mount` command lists the node name, the device name, the name under which it is mounted, the virtual-file-system type, the date and time it was mounted, and its options.

2. To mount the remote directory on to a local directory, enter:

```
mount testsys3:/test /mnt
```

This command mounts the `/test` directory located on `testsys3` onto the local `/mnt` directory.

Related Information

The `backupios` command.

mv command

Purpose

Moves files.

Syntax

To Move and Rename a File

```
mv [ -E{force|ignore|warn} ] [ -i | -f ] [ -I ] SourceFile ... TargetFile
```

To Move and Rename a Directory

```
mv -E{force|ignore|warn} [ -i | -f ] [ -I ] SourceDirectory ... TargetDirectory
```

To Move Files or Directories to a Directory Maintaining Original File Names

mv -E{force|ignore|warn}] [-i | -f] [-I] *SourceFile/SourceDirectory TargetDirectory*

Description

Attention: The **mv** command can overwrite many existing files unless you specify the **-i** flag. The **-i** flag prompts you to confirm before it overwrites a file. If both the **-f** and **-i** flags are specified in combination, the last flag specified takes precedence.

The **mv** command moves files and directories from one directory to another or renames a file or directory. If you move a file or directory to a new directory, it retains the base file name. When you move a file, all links to other files remain intact, except when you move it to a different file system. When you move a directory into an existing directory, the directory and its contents are added under the existing directory.

When you use the **mv** command to rename a file or directory, the *TargetDirectory* parameter can specify either a new file name or a new directory path name.

If moving the file would overwrite an existing file that does not have write-permission set and if standard input is a workstation, the **mv** command displays the file-permission code and reads a line from standard input. If that line begins with a *y* or the locale's equivalent of a *y*, the **mv** command moves the file. If the response is anything other than a *y*, the **mv** command does nothing to that file and continues with the next specified file. The file-permission code displayed may not fully represent the access permission if the *TargetFile* is associated with an ACL. When the parent directory of the *SourceFile* is writable and has the sticky bit set, one or more of the following conditions are true:

- The user must own the file.
- The user must own the directory
- The user must be a privileged user.
- The file must be writable by the user.

This warning message and prompt for input can be overridden by using the **-f** option.

You can use the **mv** command to move files within the same file system or between file systems. Whether you are working in one file system or across file systems, the **mv** command copies the file to the target and deletes the original file. The **mv** command preserves in the new file the time of the most recent data modification, the time of the most recent access, the user ID, the group ID, the file mode, the extended attributes, and ACLs of the original file. For symbolic links, the **mv** command preserves only the owner and group of the link itself.

If it is unable to preserve the owner and group ID, the **mv** command clears S_ISUID and S_ISGID bits in the target. The **mv** command prints a diagnostic message to stderr if it is unable to clear these bits, though the exit code is not affected.

The **mv** command modifies either the source file or the destination path if the command is prematurely terminated.

Note: The **mv** command supports the **—** (dash, dash) parameter as a delimiter that indicates the end of the flags.

The **mv** command will not move an object if the object is exported as an NFS version 4 referral. The referral object is marked as busy and remains so until it is unexported.

Flags

Attention: The **mv** command can overwrite many existing files unless you specify the **-i** flag. The **-i** flag prompts you to confirm before it overwrites a file. If both the **-f** and **-i** flags are specified in combination, the last flag specified takes precedence.

- E** The **-E** option requires one of the following arguments. If you omit the **-E** option, **warn** is the default behavior.
 - force** Fails the **mv** operation on a file if the fixed extent size or space reservation of the file cannot be preserved.
 - ignore** Ignores any errors in preserving extent attributes.
 - warn** Issues a warning if the space reservation or the fixed extent size of the file cannot be preserved.
- f** Does not prompt you before overwriting an existing file.
- i** Prompts you before moving a file or directory to an existing path name by displaying the name of the file followed by a question mark. If you answer with a line starting with **y** or the locale's equivalent of a **y**, the move continues. Any other reply prevents the move from occurring.
- I** Suppresses the warning message during ACL conversion.

Exit Status

- 0** All input files were moved successfully.
- >0** An error occurred.

Examples

1. To rename a file, enter:

```
mv appendix apndx.a
```

This command renames `appendix` to `apndx.a`. If a file named `apndx.a` already exists, its old contents are replaced with those of `appendix`.

2. To move a directory, enter:

```
mv book manual
```

This command moves all files and directories under `book` to the directory named `manual`, if `manual` exists. Otherwise, the directory `book` is renamed `manual`.

3. To move a file to another directory and give it a new name, enter:

```
mv intro manual/chap1
```

This command moves `intro` to `manual/chap1`. The name `intro` is removed from the current directory, and the same file appears as `chap1` in the directory `manual`.

4. To move a file to another directory, keeping the same name, enter:

```
mv chap3 manual
```

This command moves `chap3` to `manual/chap3`

Note: Examples 1 and 3 name two files, example 2 names two existing directories, and example 4 names a file and a directory.

5. To move several files into another directory, enter:

```
mv chap4 jim/chap5 /home/manual
```

This command moves the `chap4` file to the `/home/manual/chap4` file directory and the `jim/chap5` file to the `/home/manual/chap5` file.

6. To use the **mv** command with pattern-matching characters, enter:

```
mv manual/* .
```

This command moves all files in the `manual` directory into the current directory `.` (period), retaining the names they had in `manual`. This move also empties `manual`. You must type a space between the asterisk and the period.

Note: Pattern-matching characters expand names of existing files only. For example, the command `mv intro man*/chap1` does not work if the file `manual/chap1` does not exist.

Files

`/usr/bin/mv` Contains the `mv` command.

Related Information

The `chmod` command and the `rm` command.

netstat command

Purpose

Shows network status.

Syntax

To display active sockets for each protocol or routing table information:

```
netstat [ -num ] [ -routtable ] [ -routinfo ] [ -state ] [ -protocol Protocol ] [ Interval ]
```

To display the contents of a network data structure:

```
netstat [ -stats | -cdlistats ] [ -protocol protocol ] [ Interval ]
```

To display the address resolution protocol:

```
netstat -arp
```

To clear all statistics:

```
netstat -clear
```

To display network sockets:

```
netstat -socket
```

Description

The `netstat` command symbolically displays the contents of various network-related data structures for active connections.

Flags

<code>-arp</code>	Displays address resolution interfaces.
<code>-cdlistats</code>	Shows statistics for CDLI-based communications adapters.
<code>-clear</code>	Clears all statistics.

-num	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.
-protocol <i>protocol</i>	Shows statistics about the value specified for the <i>protocol</i> variable, which is either a well-known name for a protocol or an alias for it. A null response means that there are no numbers to report. The program report of the value specified for this variable is unknown if there is no statistics routine for it.
-routinfo	Shows the routing tables, including the user-configured and current costs of each route.
-routtable	Shows the routing tables. When used with the -stats flag, the -routtable flag shows routing statistics. See Routing Table Display.
-socket	Specifies that network sockets are to be displayed.
-state	Shows the state of all configured interfaces.
	The interface display format provides a table of cumulative statistics for the following items: <ul style="list-style-type: none"> • Errors • Collisions <p>Note: The collision count for Ethernet interfaces is not shown.</p> • Packets transferred
	The interface information that is displayed also provides the interface name, number, and address, as well as the maximum transmission units (MTUs).
-stats	Shows statistics for each protocol.
<i>Interval</i>	Continuously displays information, in seconds, regarding packet traffic on the configured network interfaces.

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. If the address can be resolved to a symbolic host name, the host address, as well as network addresses, are displayed symbolically.

NS addresses are 12-bytes in length, 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, the word and byte are reversed.

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

Interface Display

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 applicable.

- Packets transferred

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

Routing table display

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

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

The routing table contains the following fields:

Flags	The flags field in the routing table shows the state of the route: A An Active Dead Gateway Detection is enabled on 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 W The route is a cloned route 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 s The group routing stopsearch option is enabled on the route
Gateway	Direct routes are created for each interface attached to the local host.
Refs	Shows the address of the outgoing interface. Shows 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	Lists the Path Maximum Transfer Unit (PMTU).
Interface	Indicates the network interfaces utilized for the route.
Exp	Displays the time (in minutes) remaining before the route expires.
Groups	Provides a list of group IDs associated with that route.

Netmasks	Lists the netmasks applied on the system.
Route Tree for Protocol Family	Specifies the active address families for existing routes. Values for this field are as follows:
1	Specifies the UNIX address family
2	Specifies the Internet address family (for example, TCP and UDP)
3	Specifies the Xerox Network System (XNS) address family

When a value is specified for the *Interval* variable, 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 first line 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.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

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

```
netstat -routtable
```

This produces the output similar to the following:

```
Routing tables
Destination      Gateway          Flags  Refs    Use  If  PMTU Exp Groups

Route tree for Protocol Family 2 (Internet):
default          129.3.141.1     UGc    0        0  en0  -  -
129.33.140/23    127.0.0.1      U       6       53  en0  -  -
129.33.41.2      localhost      UGHS   6       115  lo0  -  -
129.45.41.2      129.3.41.1     UGHW   1       602  en0  1500 -
dcefs100         129.31.41.1    UGHW   1         2  en0  -  -
192.100.61       localhost      U       7     14446  lo0  -  -

Route tree for Protocol Family 24 (Internet v6):
::1              ::1            UH     0         0  lo0  16896 -
```

- To display interface information for an Internet interface, type:

```
netstat -state
```

This produces the output similar to the following:

```
Name  Mtu  Network  Address          Ipkts  Ierrs  Opkts  Oerrs  Coll
en0   1500  link#2   0.5.20.4.0.4e   874986  0      22494  0      0
en0   1500  90.34.14 hostname         874986  0      22494  0      0
lo0   16896 link#1                    14581  0      14590  0      0
lo0   16896 129      localhost        14581  0      14590  0      0
lo0   16896 ::1                    14581  0      14590  0      0
```

- To display network sockets:

```
netstat -socket
```

Related Information

The **mktcpip** command, the **hostname** command, the **startnetsvc** command, the **stopnetsvc** command, the **cfglnagg** command, the **entstat** command, the **cfgnamesrv** command, the **hostmap** command, the **traceroute** command, the **ping** command, and the **optimizenet** command.

oem_platform_level command

Purpose

Returns the operating system level of the OEM installation and setup environment.

Syntax

```
oem_platform_level
```

Description

The **oem_platform_level** command displays the name and version of the underlying Virtual I/O Server operating system.

This command can only be executed by the prime administrator.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1

Examples

To get the operating system level of the OEM installation and setup environment, run the following command:

```
oem_platform_level
```

Related Information

The **lssw** command, the **ioslevel** command, the **remote_management** command, the **oem_setup_env** command, and the **updateios** command.

oem_setup_env command

Purpose

Initiates the OEM installation and setup environment.

Syntax

```
oem_setup_env
```

Description

The **oem_setup_env** command places the user into the OEM software installation and setup environment. In this environment, the user can install and setup OEM software by following the installation instructions provided with each software package. After the software is installed, the user will need to create a link in the **/usr/ios/oem/** directory to any new commands that will run from the Virtual I/O Server command line. After these links have been created, the commands will be accessible by all Virtual I/O Server users. Note however that these commands will not run with root authority.

After the software has been installed,, typing **exit** will return the user to the Virtual I/O Server prompt.

Only the prime administrator can execute this command.

Note: The `oem_setup_env` command will place the **padmin** user in a non-restricted UNIX root shell with a home directory in the `/home/padmin` directory. The user can then run any command available to the root user. This is not a supported Virtual I/O Server administration method. The purpose of this command is to allow installation of vendor software, such as device drivers.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1

Examples

To initiate the OEM setup and installation environment, type the following:

```
oem_setup_env
```

Related Information

The `lssw` command, the `ioslevel` command, the `remote_management` command, the `updateios` command, and the `oem_platform_level` command.

optimizenet command

Purpose

Manages network tuning parameters.

Syntax

```
optimizenet [ -reboot | -perm ] { -set Tunable[=NewValue] | -default Tunable }
```

```
optimizenet -list [ Tunable ]
```

```
optimizenet -h [ Tunable ]
```

Description

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

If the **-list** flag is specified without any *Tunables*, only *Tunables* modifiable by this command will be displayed.

Flags

-default <i>Tunable</i>	Resets <i>Tunable</i> to its default value. If the <i>Tunable</i> needs to be changed (not currently set to its default value) and it is of type Reboot, it will not be changed; a warning displays instead.
-h <i>Tunable</i>	Displays help about <i>Tunable</i> parameter if one is specified.

-list [Tunable]

Lists the characteristics of one or all Tunables, one per line, using the following format:

NAME	CUR	DEF	BOOT	MIN	MAX	UNIT	TYPE
DEPENDENCIES							

General Network Parameters							

sockthresh	85	85	85	0	100	%_of_thewall	D

fasttimo	200	200	200	50	200	millisecond	D

inet_stack_size	16	16	16	1		kbyte	R

where

CUR current value

DEF default value

BOOT reboot value

MIN minimal value

MAX maximum value

UNIT tunable unit of measure

TYPE Parameter type: D (Dynamic), S (Static), R (Reboot), B (Bosboot), M (Mount), I (Incremental), and C (Connect)

DEPENDENCIES

List of dependent tunable parameters, one per line

-perm

Makes changes apply to both current and reboot values when used in combination with **-set** or **-default**. These combinations cannot be used on Reboot type parameters because their current value can't be changed.

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

-reboot

Makes changes apply to reboot values when used in combination with **-set** or **-default**.

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

-set Tunable
[=*NewValue*]

Displays the value or sets the Tunable to *NewValue*. If a tunable needs to be changed (the specified value is different than current value), and is of type **Reboot** it will not be changed but a warning displays instead.

When **-reboot** is used in combination without a new value, the **nextboot** value for Tunable is displayed. When **-perm** is used in combination without a new value, a value displays only if the current and next boot values for tunable are the same Otherwise NONE displays as the value.

Network tunable parameters

arptab_bsiz	<p>Purpose: Specifies Address Resolution Protocol (ARP) table bucket size.</p> <p>Values: Default: 7 Range: 1 to MAXSHORT Type: Reboot</p> <p>Diagnosis netstat -protocol arp will show the number of ARP packets sent and the number of ARP entries purged from the ARP table. If large number of entries are being purged, the ARP table size should be increased.</p>
arptab_nb	<p>Purpose: Specifies the number of ARP table buckets.</p> <p>Values: Default: 73 Range: 1 to MAXSHORT Type: Reboot</p> <p>Diagnosis: netstat -protocol arp will show the number of ARP packets sent and the number of ARP entries purged from the ARP table. If large number of entries are being purged, the ARP table size should be increased.</p> <p>Increase this value for systems that have a large number of clients or servers. The default provides for $73 \times 7 = 511$ ARP entries, but assumes an even hash distribution.</p>
clean_partial_conns	<p>Purpose: Specifies whether or not SYN (synchronizes the sequence number) attacks are being avoided.</p> <p>Values: Default: 0 (off) Range: 0 or 1 Type: Dynamic</p> <p>Tuning: This option should be turned on for servers that need to protect against network attacks. If on, randomly removes partial connections to make room for new non-attack connections.</p>

net_malloc_police	<p>Purpose: Specifies the size of the net_malloc and net_free trace buffers.</p> <p>Values: Default: 0 Range: 0 to MAXINT Type: Dynamic</p> <p>Tuning: If the value of this variable is non-zero, all net_malloc and net_free buffers will be traced in a kernel buffer and by system trace hook HKWD_NET_MALLOC. Additional error-checking will also be enabled. This includes checks for freeing a free buffer, alignment, and buffer overwrite. Enable this parameter only when investigating some network problem, because performance is affected negatively when turned on. The default value is zero (policing off). Values of net_malloc_police larger than 1024 will allocate that many items in the kernel buffer for tracing.</p>
rfc1323	<p>Purpose: Enables window scaling and timestamps as specified by RFC 1323 (TCP Extensions for High Performance). Window scaling allows the TCP window sizes (tcp_recvspace and tcp_sendspace) to be larger than 64KB (65536) and is typically used for large MTU networks.</p> <p>Values: Default: 0 (off) Range: 0 or 1 Type: Connect</p> <p>Tuning: The default value of 0 disables the RFC enhancements on a systemwide scale. A value of 1 specifies that all TCP connections will attempt to negotiate the RFC enhancements. Make changes before attempting to set tcp_sendspace and tcp_recvspace to more than 64 KB.</p>
route_expire	<p>Purpose: Specifies whether unused routes created by cloning, or created and modified by redirects expire.</p> <p>Values: Default: 1 (on) Range: 0 or 1 Type: Dynamic</p> <p>Tuning: A value of 1 allows route expiration, which is the default. Negative values are not allowed for this option.</p>
tcp_pmtu_discover	<p>Purpose: Enables or disables path MTU discovery for TCP applications.</p> <p>Values: Default: 1 Range: 0 or 1 Type: Dynamic</p> <p>Tuning: A value of 0 disables path MTU discovery for TCP applications, while a value of 1 enables it.</p>

tcp_recvspace	<p>Purpose: Specifies the system default socket buffer size for receiving data. This affects the window size used by TCP.</p> <p>Values: Default: 16384 bytes Range: 4096 to 1048576 Type: Connect</p> <p>Diagnosis: Setting the socket buffer size to 16 KB (16,384) improves performance over standard Ethernet and Token-Ring networks. Lower bandwidth networks, such as Serial Line Internet Protocol (SLIP), or higher bandwidth networks, such as Serial Optical Link, should have different optimum buffer sizes. The optimum buffer size is the product of the media bandwidth and the average round-trip time of a packet. For high-speed networks, like gigabit Ethernet or ATM 622, a value of 65536 should be used for the minimum size for best performance.</p> <p>For values larger than 65536, you must enable rfc1323 (rfc1323=1) to enable TCP window scaling.</p>
tcp_sendspace	<p>Purpose: Specifies the system default socket buffer size for sending data.</p> <p>Values: Default: 16384 bytes Range: 4096 to 1048576 Type: Connect</p> <p>Tuning: This affects the window size used by TCP. Setting the socket buffer size to 16 KB (16,384) improves performance over standard Ethernet networks. Lower bandwidth networks, such as Serial Line Internet Protocol (SLIP), or higher bandwidth networks, such as Serial Optical Link, should have different optimum buffer sizes. The optimum buffer size is the product of the media bandwidth and the average round-trip time of a packet: (optimum_window=bandwidth * average_round_trip_time) For high-speed networks, like gigabit Ethernet or ATM 622, a value of 65536 should be used for the minimum size for best performance. For values larger than 65536, you must enable rfc1323 (rfc1323=1) to enable TCP window scaling.</p>
udp_recvspace	<p>Purpose: Specifies the system default socket-buffer size for receiving UDP data.</p> <p>Values: Default: 42080 bytes Range: 4096 to 1048576 Type: Connect</p> <p>Diagnosis: Nonzero n in netstat -stats report of udp: n socket buffer overflows</p> <p>Tuning: Increase size, preferably to multiple of 4096.</p>

udp_sendspace	<p>Purpose: Specifies the system default socket-buffer size for sending UDP data.</p> <p>Values: Default: 9216 bytes Range: 4096 to 1048576 Type: Connect</p> <p>Diagnosis: Increase size, preferably to multiple of 4096.</p>
----------------------	---

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

1. To display the maximum size of the mbuf pool, type:
`optimizenet -set thewall`
2. To change the default socket buffer sizes on your system, type:
`optimizenet -reboot -set tcp_sendspace=32768`
3. To use a machine as an internet work router over TCP/IP networks, type:
`optimizenet -set ipforwarding=1`
4. To list the current and reboot value, range, unit, type and dependencies of the **arptab_bsiz** parameter, type:
`optimizenet -list arptab_bsiz`
5. To display help information on **arptab_bsiz**, type:
`optimizenet -h arptab_bsiz`

Related Information

The `entstat` command, the `lsnetsvc` command, the `mktcpip` command, the `netstat` command, and the `traceroute` command.

IVM `os_install` Command

Purpose

Performs network install operations on **OS_install** objects.

Syntax

Traditional usage:

```
OS_install {-o Operation} [ -a attr=value... ] {ObjectName}
```

For system plan installations (System Plan mode):

```
OS_install -i sysplan {-x sysplan.xml} [ -d ]
```

For listing OS_install objects (List mode):

```
OS_install -l [-v] [-t object_type | object_name]
```


Description

The **OS_install** command performs a network installation operation on an **OS_install** object. The type of operation is dependent on the type of object specified by the ObjectName parameter. The object pointed to by the ObjectName parameter can be one of three types: client, **OS_Resource**, or **ControlHost**. Command operations involve the creation and management of **OS_install** objects in order to network install an operation system onto a client machine.

OS_install can also be run in System Plan mode by passing the **-i sysplan** flag instead of specifying an operation. This provides the ability to combine multiple OS_install operations into a single XML document.

The List mode of **OS_install** is used to list the current configuration of objects in the **OS_install** environment.

Flags

-a <i>attr=value</i>	Assigns the specified value to the specified attribute. “Operations” on page 314 lists the required and optional attributes for a specific operation.
-d	Destroys all OS_install objects created during System Plan mode after all operations have been completed.
-i sysplan	Specifies System Plan mode.
-l	Lists all OS_install objects in the environment by default.
-o <i>Operation</i>	Specifies an operation to perform on an OS_install object.
-t <i>object_type</i> <i>object_name</i>	Narrows the list returned by the -l flag to only objects of type <i>object_type</i> or to the single OS_install object specified by <i>object_name</i> .
-v	Displays the list returned by the -l flag.
-x <i>sysplan.xml</i>	Specifies the XML file that contains the system plan.

Operations

Operation	Description	Required Attributes	Optional Attributes
define_client [-a <i>attr=value...</i>] { <i>ClientObjectName</i> }	Defines a new client object.	ip_addr Client's IP address. mac_addr MAC address of client's network interface. gateway Client's IP gateway. subnet_mask Client's IP subnet mask.	adapter_speed Speed of client's network adapter. adapter_duplex Duplex setting of client's network adapter. lpar LPAR name to install client. profile LPAR profile to use for client. managed_system Name of managed system that contains LPAR. disk_location Location of disk to install client. ctrl_host Name of Hardware Control Host object for this client.
define_resource [-a <i>attr=value...</i>] { <i>ResourceObjectName</i> }	Defines a new OS_Resource object.	type AIX or Linux version Operating system version	location Absolute path where OS_Resource will reside. source Source of installation images. configfile Installation configuration file.
define_ctrl_host [-a <i>attr=value...</i>] { <i>ControlHostObjectName</i> }	Defines a new Hardware Control Host object.	communication_method ssh, rsh, or local hostname Host name of control host. type HMC or ivm	None.

Operation	Description	Required Attributes	Optional Attributes
allocate [-a <i>attr=value...</i>] { <i>ClientObjectName</i> }	Allocates an OS_Resource to a client object. Both objects must already exist in the OS_install environment. An error will occur if the client object has an OS_Resource already allocated to it.	os_resource Existing OS_Resource object to allocate to the client object.	None.
netboot { <i>ClientObjectName</i> }	Instructs the hardware control host of the client object to initiate a network boot.	None.	None.
monitor_installation { <i>ClientObjectName</i> }	Monitors the installation status of the client object.	None.	None.
deallocate { <i>ClientObjectName</i> }	Deallocates the OS_Resource that was allocated to the client object by an allocate operation.	None.	None.
remove { <i>ObjectName</i> }	Removes the object from the OS_install environment.	None.	None.

Exit Status

The following exit values are returned:

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

Examples

1. To define a client object, enter a command similar to the following:

```
OS_install -o define_client -a ip_addr=128.0.64.117
-a mac_addr=ab:cc:de:10:23:45 -a gateway=128.0.64.1
-a subnet_mask=255.255.255.0 -a ctrl_host=myhmc -a lpar=AIX1
-a profile=AIX1 -a managed_system=myMngSys myclient01
```

2. To define an **OS_Resource** object, enter a command similar to the following:

```
OS_install -o define_resource -a location=/images/AIX/53ML3 -a type=AIX
-a version=53ML3 my53resource
```

3. To allocate the **OS_Resource** object defined in the preceding example to a client object, enter a command similar to the following:

```
OS_install -o allocate -a os_resource=my53resource myclient01
```

4. To deallocate the my53resource client object that was allocated in the preceding example, enter:

```
OS_install -o deallocate myclient01
```

5. To define a **ControlHost** object to be specified for the **ctrl_host** attribute, enter a command similar to the following:

```
OS_install -o define_ctrl_host -a type=ivm -a hostname=ivm_hostname
-a communication_method=ssh myivm
```

6. To view a myclient01 installation, enter:

```
OS_install -o monitor_installation myclient01
```

7. To remove the definition of the my53resource object, enter:

```
OS_install -o remove my53resource
```

8. To remove the definition of the myclient01 object, enter:

```
OS_install -o remove myclient01
```

Note: If an **OS_Resource** object is specified, the remove operation also removes any OS images that exist in the file system directory specified by the object's location attribute.

Related Information

The `lparnetboot` command.

passwd command

Purpose

Changes a user's password.

Syntax

```
passwd [ User ]
```

Description

The **passwd** command sets and changes passwords for users. Use this command to change your own password (all users) or another user's password (padmin only). To change your own password, enter the **passwd** command. The **passwd** command prompts the non-padmin user for the old password and then prompts for the new password twice. The password is never displayed on the screen. If the two entries of the new password do not match, the **passwd** command prompts for the new password again.

Password policy is checked during a password change. Construct locally defined passwords according to the following password restrictions:

minother	Specifies the minimum number of other characters.
minlen	Specifies the minimum number of characters.
maxrepeats	Specifies the maximum number of times a single character can be used in a password.
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.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Security

Changing a password other than your own requires prime administrator authority.

Examples

1. To change the password for user account **heerlen**, type:
passwd heerlen

The user will then be prompted to enter the new password.

Related Information

The chuser command, the lsuser command, the mkuser command, and the rmuser command.

pdump command

Purpose

Perform platform (hardware and firmware) dump-related actions.

Syntax

```
pdump -reserve fstype | -enable | -disable | -ls | -size
```

Description

The **pdump** command helps the operating system save firmware and hardware related dumps. This command also provides an estimate of the disk space required for storing these dumps. Note that platform and scan dumps are saved in order to capture the state of the firmware and the hardware for analysis.

Flags

-disable	Disables platform dumps.
-enable	Enables platform dumps.
-ls	Lists the current configuration of platform dump.
-reserve <i>fstype</i>	Reserves enough disk space on the system for platform dumps. The -enable option will create a file system (if one does not exist) exclusively for platform dumps. If a file system already exists and the size is not enough, the file system size will be increased. The <i>fstype</i> must be a valid file system type. If the file system already exists, any may be specified.
-size	Provides an estimate of disk space required to save the platform dumps when they occur. This option will interact with the firmware to provide this estimate. It is expected that, based on this space information, the user will have enough disk space allocated for platform dumps to be saved. The value output will be the required size in bytes.

ping command

Purpose

Sends an echo request to a network host.

Syntax

```
ping [ -n ] [ -r ] [ -s PacketSize ] [ -src hostname/IP_addr ] Host [ Count ]
```

Description

The **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). Because of the load that continuous echo requests can place on the system, repeated requests should be used primarily for problem isolation.

Flags

-n	Specifies numeric output only. No attempt is made to look up symbolic names for host addresses.
-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.
-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.
-src hostname/IP_addr	Uses the IP address as the source address in outgoing ping packets. On hosts with more than one IP address, the -src flag can be used to force the source address to be something other than the IP address of the interface on which the packet is sent. If the IP address is not one of the machine's interface addresses, an error is returned and nothing is sent.

Parameters

<i>Count</i>	Specifies the number of echo requests to be sent (and received). This parameter is included for compatibility with previous versions of the ping command.
--------------	--

Exit Status

See "Exit status for Virtual I/O Server commands" on page 1.

Examples

1. To check the network connection to host **canopus** and specify the number of echo requests to send, enter:

```
ping canopus 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 obtain information about host **opus** and specify the number of data bytes to be sent, enter:

```
ping -s 2000 opus
```

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.

Related Information

The `optimizenet` command and the `traceroute` command.

prepdev command

Purpose

Prepares a device to be assigned to a shared memory pool (to be used as a paging space device by a shared memory partition).

Syntax

```
prepdev { -dev devicename }
```

Description

The **prepdev** command prepares a device to be assigned to a shared memory pool (to be used as a paging space device by a shared memory partition). This command checks if the device you specify using the **-dev** flag is already assigned or used by the following:

- Assigned to a shared memory pool (being used as a paging device)
- Used as a backup device for virtual SCSI
- Used by a file system, logical volume, or volume group.

If the device is already being used or assigned to a shared memory pool, you receive system output that tells you how to prepare the device to be used as a paging space device.

Flags

<code>-dev <i>devicename</i></code>	Specifies the logical device to be used as a paging space device.
<code>--help</code>	Displays the help text for this command and exit.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Example

1. To verify that `hdisk5` is ready to be used as a paging space device by a shared memory partition, type the following command:

```
prepdev -dev hdisk5
```

Related Information

The `lspv` command and the `rmvdev` command.

redefvg command

Purpose

Redefines the set of physical volumes of the given volume group in the device configuration database.

Syntax

```
redefvg { -dev Device | -vgid 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 `redefvg` command determines which physical volumes belong to the specified volume group and enters this information in the device configuration database. The `redefvg` command checks for inconsistencies by reading the reserved areas of all the configured physical volumes attached to the system.

Flags

<code>-dev <i>Device</i></code>	The volume group ID, <i>Vgid</i> , is read from the specified physical volume device. You can specify the <i>Vgid</i> of any physical volume belonging to the volume group that you are redefining.
<code>-vgid <i>Vgid</i></code>	The volume group identification number of the volume group to be redefined.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

1. To synchronize the copies on physical volumes `hdisk04`, type:

```
redefvg -dev hdisk04
```
2. To synchronize the copies on volume groups `vg04` and `vg05`, type:

```
redefvg -vgid vg04 vg05
```


Related Information

The **mkvg** command, the **syncvg** command, the **chvg** command, the **extendvg** command, the **lsvg** command, the **mirrorios** command, the **unmirrorios** command, the **activatevg** command, the **deactivatevg** command, the **importvg** command, the **exportvg** command, and the **reducevg** 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 [ -f ] [ -rmlv ]VolumeGroup PhysicalVolume ...
```

Description

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.

Note:

- Sometimes a disk is removed from the system without first running the **reducevg** command. The VGDA still has this removed disk in its memory, but the *PhysicalVolume* name no longer exists or has been reassigned. To remove references to this missing disk you can still use **reducevg**, but with the Physical Volume ID (PVID) instead of the disk name: `reducevg VolumeGroup PVID`.
- The specified physical volume cannot contain a logical volume that is assigned to a shared memory pool (to be used as a paging space device by a shared memory partition).

Flags

-f	Removes the requirement for user confirmation when the -rmlv flag is used.
-rmlv	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 **-rmlv** 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.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

1. To remove physical volume **hdisk1** from volume group **vg01**, type:
`reducevg vg01 hdisk1`
2. To remove physical volume **hdisk1** and all residing logical volumes from volume group **vg01** without user confirmation, type:
Attention: The **reducevg** command with the **-rmlv** flag automatically deletes all logical volume data before removing the physical volume.

```
reducevg -rmlv -f vg01 hdisk1
```

The physical volume **hdisk1** and all residing logical volumes are removed.

Related Information

The **mkvg** command, the **chvg** command, the **extendvg** command, the **lsvg** command, the **mirrorios** command, the **unmirrorios** command, the **activatevg** command, the **deactivatevg** command, the **importvg** command, the **exportvg** command, and the **syncvg** command.

remote_management command

Purpose

Enables the Virtual I/O Server to be remotely managed by an AIX NIM (Network Installation Management) master.

Syntax

To enable the Virtual I/O Server to be remotely managed by an AIX NIM master:

```
remote_management [ -interface Interface ] Master
```

To disable remote management:

```
remote_management -disable
```

Description

The **remote_management** command will setup the Virtual I/O Server to allow remote management from a NIM master. The *Master* parameter specifies the NIM master *hostname*. The *Interface* parameter specifies the network interface to be used to connect to the NIM master. If *Interface* is not specified, the default network interface used will be **en0**.

The **remote_management** command utilizes the NIM service handler for client communication (nimsh), so the NIM master must be nimsh capable.

Once remote management has been enabled on the Virtual I/O Server, typical NIM functions, such as update, backup, and reinstall, can be initiated from the NIM master.

Flags

-disable	Disables the Virtual I/O Server NIM client daemon.
-interface	Specifies which network interface to use. If no network interface is specified, interface en0 will be used.

Exit Status

See "Exit status for Virtual I/O Server commands" on page 1

Examples

1. To enable **remote_management** using NIM master **nimsys01**, type:

```
remote_management nimsys01
```
2. To disable **remote_management**, type:

```
remote_management -disable
```

Related Information

The `lssw` command, the `ioslevel` command, the `updateios` command, the `oem_setup_env` command, and the `oem_platform_level` command.

replphyvol command

Purpose

Replaces a physical volume in a volume group with another physical volume.

Syntax

```
replphyvol [-force ] {SourcePhysicalVolume | SourcePhysicalVolumeID} [DestinationPhysicalVolume]
```

```
replphyvol [-retry] dir_name [DestinationPhysicalVolume]
```

Description

The **replphyvol** 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 **replphyvol** command cannot replace a *SourcePhysicalVolume* with stale logical volume unless this logical volume has a non-stale mirror.
3. The **replphyvol** command is not allowed on a snapshot volume group or a volume group that has a snapshot volume group. The allocation of the new physical partitions follows the policies defined for the logical volumes that contain the physical partitions being replaced.
4. The *DestinationPhysicalVolume* cannot be a physical volume that is assigned to a shared memory pool (to be used as a paging space device by a shared memory partition).

The allocation of the new physical partitions follows the policies defined for the logical volumes that contain the physical partitions being replaced.

Flags

-force	Runs the command without prompting the user to continue.
-retry	Displays specified boot list, after any specified alterations
<i>SourcePhysicalVolume</i>	Specifies the source physical volume name.
<i>SourcePhysicalVolumeID</i>	Specifies the source physical volume ID.
<i>DestinationPhysicalVolume</i>	Specifies the destination physical volume name.
<i>dir_name</i>	Recovers replphyvol if it is interrupted by CTL+C, a system lock up, or a loss of quorum. Specify the directory name given during the initial run of replphyvol if you are using the <code>-retry</code> flag. Using this flag, you can also change the <i>DestinationPhysicalVolume</i>

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

To create a boot list of logical devices to be used in the next normal boot, type the following command:

```
replphyvol -force hdisk0 hdisk4
```

To recover a boot list of logical devices if the normal boot was interrupted, type the following command:

```
replphyvol -retry <directory_name> hdisk4
```

Related Information

The **migratepv** command.

restore command

Purpose

Extracts files from archives created with the **backup** command.

Syntax

To restore files archived by file name:

```
restore -x [ d M n Q v q e ] [ -b Number ] [ -f Device ] [ -s SeekBackup ] [ -E { force | ignore | warn } ] [ File ... ]
```

To list files archived by file name:

```
restore -T | -t [ a l n q v Q ] [ -b Number ] [ -f Device ] [ -s SeekBackup ]
```

To restore files archived by file system:

- **restore -r** [**B n q v y**] [**-b Number**] [**-f Device**] [**-s SeekBackup**]
- **restore -R** [**B n v y**] [**-b Number**] [**-f Device**] [**-s SeekBackup**]
- **restore -i** [**h m n q v y**] [**-b Number**] [**-f Device**] [**-s SeekBackup**]
- **restore -x** [**B h n m q v y**] [**-b Number**] [**-f Device**] [**-s SeekBackup**] [*File ...*]

To restore files beginning at a specified volume number:

```
restore -X Number [ -Md n q v e Q ] [ -b Number ] [ -f Device ] [ -s Number ] [ -E { force | ignore | warn } ] [ File ... ]
```

To list files archived by file system:

```
restore -t | -T [ B a l n h q v y ] [ -b Number ] [ -f Device ] [ -s SeekBackup ] [ File ... ]
```

To restore file attributes archived by file name:

```
restore -Pstring [ B d q v Q ] [ b Number ] [ s SeekNumber ] [ -f Device ] [ File ... ]
```

To restore file attributes archived by file system:

```
restore -Pstring [ h q v ] [ b Number ] [ s SeekNumber ] [ -f Device ] [ File ... ]
```

Description

The **restore** command reads archives created by the backup operation 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: The **restore** command determines the actively sparses files that are being restored. If a file has block aligned and sized areas that are NULL populated, then the **restore** command does not cause physical space for those file system blocks to be allocated. The size in bytes of the file remain the same, but the actual space taken within the file system 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`.

Note:

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 was created using the **backup** command and was made to a tape device with the device block size set to 0, it might be necessary for you to have explicit knowledge of the block size that was used when the tape was created to restore from the tape.
3. Multiple archives can exist on a single tape. When you are 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 the **-B**, **-s**, or **-X** flag is specified.

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 restore operation.

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 by using the **backup** command. The restore operation restores the files from a file name archive without any special understanding of the underlying structure of the file system. The restore operation 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 file name should be enclosed in single quotations, and patterns should be enclosed in parentheses (...).

About sparse files

File system files for the operating system that contain long strings of NULLs can be stored much more efficiently than other files. To be specific, if a string of NULLs spans an entire allocation block, that whole block is not stored on disk at all. Files where one or more blocks are omitted in this way are called sparse files. The missing blocks are also known as holes.

Note: Sparse files are not the same as compressed files. Sparse files are exactly the same as their nonsparse equivalents when they are read.

Sparse files are generally created by database applications. Whenever a database file is created, it is formatted with NULLs. These fragments might be either filled with allocated or unallocated NULLs.

Flags

- a** Displays the **-a** option displays the list of files in the archive, along with their permissions when specified with the **-t** or **-T** flag.
- B** Specifies that the archive should be read from standard input. Normally, the restore operation 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** operation 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** operation 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** operation 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** operation writes an error message to standard error and exits with a nonzero return code. Larger values for the *Number* parameter result in larger physical transfers from the tape device.

The value of the **-b** flag is always ignored when the **restore** command reads from diskette. In this case, the command always reads in clusters that occupy a complete track.
- d** Indicates that, if the *File* parameter is a directory, all files in that directory should be restored. This flag can only be used when the archive is in file name format.

-e

Restores the nonsparse files because they were archived by the name format of the backup operation for both packed and unpacked files. It is necessary to know the sparseness or nonsparseness of the files before archiving the files, since enabling this flag restores the sparse files as nonsparse.

This flag should be enabled only if the files being restored are nonsparsed, consisting of more than 4K NULLs. If the **-e** flag is specified during the restore operation, it successfully restores all normal files normally and nonsparse database files as nonsparse.

-E

Extracts beginning at a specified volume number and requires one of the following arguments. If you omit the **-E** flag, **warn** is the default behavior.

force Fails the restore operation on a file if the fixed extent size or space reservation of the file cannot be preserved.

ignore Ignores any errors in preserving extent attributes.

warn Issues a warning if the space reservation or the fixed size of the file cannot be preserved.

-f Device

Specifies the input device. To receive input from a named device, specify the *Device* variable as a path name (such as */dev/rmt0*). To receive input from the standard output device, specify a dash (-)

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 8 mm, 2.3 GB tapes or a set of 1.44 MB diskettes. All tape devices must be set to the same physical tape block size.

If the *Device* variable specifies a range, the restore operation automatically goes from one device in the range to the next. After exhausting all of the specified devices, the restore operation 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** flag.

- i Allows you to interactively restore selected files from a file system archive. The subcommands for the **-i** flag are:
 - 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.
 - cd** *Directory*: Changes the current directory to the specified directory.
 - 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).
 - extract**: Restores all the directories and files on the extraction list.
 - help**: Displays a summary of the subcommands.
 - 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 asterick (*) 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 .
 - pwd**: Displays the full path name of the current directory.
 - quit**: Causes the restore operation to exit immediately. Any files on the extraction list are not restored.
 - setmodes**: Sets the owner, mode, and time for all directories added to the extraction list.
 - 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.
- l Displays a detailed list of files, which includes the timestamp, file permissions, file size, owner, and group when specified with the **-t** or **-T** flag.. The **-l** option overrides the **-a** option.
- M Sets the access and modification times of restored files 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 operation maintains the access and modification times as they appear on the backup medium.
- 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.
- n By default the restore operation restores any Access Control List (ACL)s, Printing Color Files (PCL)s, or named extended attributes in the archive. The **-n** flag causes the restore operation to skip over any ACLs, PCLs, or named extended attributes in the archive and not restore them.

- Pstring** Restore only the file attributes. This flag does not restore the file contents. If the file specified does not exist in the target directory path, the file is not created. This flag restores file attributes selectively depending on the flags specified in the string parameter. The *String* parameter can be a combination of the following characters:
- A** Restore all attributes.
 - a** Restore only the permissions of the files.
 - o** Restore only the ownership of the files.
 - t** Restore only the timestamp of the files.
 - c** Restore only the ACL attributes of the files
- Note:** Among the existing flags for the **restore** command, flags **v**, **h**, **b**, **s**, **f**, **B**, **d**, and **q** are valid with the **P** flag. The **P** flag can be used with both file name and file system archives. If the *File* parameter is a symbolic link, then the metadata of the target file is modified and not that of the symbolic link.
- Attention:** Usage of **-P** flag overwrites the attributes of files owned by another user when run by the superuser.
- Q** For backups done by name, specifies that the command should exit upon encountering any type of error rather than attempt to recover and continue processing the archive.
- q** Specifies that the first volume is ready to use and that the restore operation should not prompt you to mount the volume and press 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 the restore operation to pass information between incremental restore operations. This file should be removed after 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 operation 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 error. This flag can be used to determine if the archive is in file name or file system format.
- T** Displays information about the backup archive. If the archive is in file name format, the information contained in the volume header is written to standard error, and a list of files found on the archive is written to standard output. The *File* parameter is ignored for file name archives. If the archive is in file system format, the behavior is identical to the **-t** flag.

-v	Displays additional information when restoring. If the archive is in file name format and either the -x or -T flag is specified, the size of the file as it exists on the archive is displayed in bytes. Directory, block, or character device files are archived with a size of 0. Symbolic links are listed with the size of the symbolic link. Hard links are listed with the size of the file, which is how they are archived. Once the archive is read, a total of these sizes is displayed. If the archive is in file system format, directory and nondirectory archive members are distinguished.
-x	<p>Restores individually named files specified by the <i>File</i> parameter. If the <i>File</i> parameter is not specified, all the archive members are restored. If the <i>File</i> parameter is a directory and the archive is in file name format, only the directory is restored. If the <i>File</i> 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 <i>File</i> parameter must be the same as the names shown by the restore -T flag. 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.</p> <p>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.</p> <p>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:</p> <p>asterick* (*) Matches zero or more characters, but not a . (period) or / (slash).</p> <p>question mark (?) Matches any single character, but not a . (period) or / (slash).</p> <p>brackets ([]) Matches any one of the characters enclosed within the brackets. If a pair of characters separated by a dash are contained within the brackets, the pattern matches any character that lexically falls between the two characters in the current local. Additionally, a . (period) or a / (slash) within the brackets will not match a . (period) or a / (slash) in a file name.</p> <p>backslash (\) Matches the immediately following character, preventing its possible interpretation as a metacharacter.</p>
-X <i>VolumeNumber</i>	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 only to file name archives.
-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 zeros. 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 diskette device `/dev/rfd0`, type the following command:

```
restore -Tq
```

The archive is read from the `/dev/rfd0` default restore device. The names of all the files and directories contained in the archive are displayed. For file system archives, the file name is preceded by the i-node number of the file as it exists on the archive. The `-q` flag tells the `restore` command that the first volume is available and is ready to be read. As a result, you are not prompted to mount the first volume.

2. To restore a specific file, type the following command:

```
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 file from tape into the virtual media repository, type the following command:

```
restore -xvqf /dev/rmt0 /var/vio/VMLibrary/media_file
```

4. To restore a specific directory and the contents of that directory from a file name archive, type the following command:

```
restore -xdvqf /dev/rmt0 /home/mike/tools
```

The `-x` flag extracts files by their file name. The `-d` flag tells the `restore` command 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.

5. To restore a specific directory and the contents of that directory from a file system archive, type the following command:

```
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.

6. To restore an entire file system archive, type the following command:

```
restore -rvqf /dev/rmt0
```

This command restores the entire file system archived on tape device `/dev/rmt0` 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).

7. To restore the fifth and ninth backups from a single-volume, multiple-backup tape, type the following command:

```
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 that 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.

8. To restore the fourth backup, which begins on the sixth tape on a 10-tape multiple-backup archive, put the sixth tape into the tape drive and type the following command:

```
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.

9. To improve the performance on streaming tape devices, pipe the **dd** command to the **restore** command and type the following command:

```
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.

10. To improve the performance of the **restore** command on 9348 Magnetic Tape Unit Model 12, you can change the block size by typing the following command:

```
chdev -l DeviceName -a BlockSize=32k
```

11. To restore nonparse database files, type the following command:

```
restore -xef /dev/rmt0
```

12. To restore files that were archived as sparse, type the following command:

```
restore -xf /dev/rmt0
```

13. To restore only the permissions of the files from the archive, type the following command:

```
restore -Pa -vf /dev/rmt0
```

14. To restore only the ACL attributes of the files from the archive, type the following command:

```
restore -Pc -vf /dev/rmt0
```

15. To view the table of contents along with the file permissions, type the following command:

```
restore -Ta -vf /dev/rmt0
```

16. To view the table of contents of a file name archive along with the timestamps and file permissions, type the following command:

```
restore -Tl -vf /dev/rmt0
```

17. To view the table of contents of a file system archive along with the timestamps and file permissions, type the following command:

```
restore -tl -vf /dev/rmt0
```

Related Information

The **backup** command.

restorevgstruct command

Purpose

Restores the user volume group.

Syntax

```
restorevgstruct { -ls | -vg VolumeGroupLabel [ DiskName ... ] }
```

Description

The **restorevgstruct** command restores the structure of a previously saved user volume group. If the **-ls** flag is specified, a list of previously-saved volume groups and the date each volume group was saved is displayed. This command does not work on rootvg.

Flags

DiskName...	Specifies the names of disk devices to be used instead of the disk devices saved in the volume group structure. 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 belong to a volume group, they must be removed from the volume group using the reducevg command. Also, the target disk device cannot be a physical volume that is assigned to a shared memory pool (to be used as a paging space device by a shared memory partition).
-ls	Displays a list of previously saved volume groups.
-vg	Specifies the name of the VolumeGroup to restore.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

- To restore the volume group **myvg**, onto the **hdisk2** and **hdisk3** disks, enter:

```
restorevgstruct myvg hdisk2 hdisk3
```
- To list all previously saved volume groups, enter:

```
restorvgstruct -ls
```

The message generated would be similar to:

```
-rw-r--r-- 1 root    system    51200 Jun 18 10:53 myvg.data
-rw-r--r-- 1 root    system    51200 Jun 18 10:53 myvg2.data
```

Related Information

The **activatevg** command, the **savevgstruct** command, the **chvg** command, the **deactivatevg** command, the **exportvg** command, the **extendvg** command, the **importvg** command, the **reducevg** command, the **lsvg** command, the **mkvg** command, and the **syncvg** command.

rm command

Purpose

Removes (unlinks) files or directories.

Syntax

```
rm [ -f ] [ -r ] [ -R ] [ -i ] [ -e ] File ...
```

Description

The **rm** command removes the entries for the specified *File* parameter from a directory. If an entry is the last link to a file, the file is then deleted. If you do not have write permission for a file and the standard input is a terminal, you are prompted with the file name and ask to confirm that you want to delete the file. If you type a *y* (for yes), the file is deleted, type any other character and the file is not deleted. You do not need read or write permission for the file you want to remove. However, you must have write permission for the directory containing the file.

If the file is a symbolic link, the link is removed, but the file or directory that the symbolic link refers to remains. You do not need write permission to delete a symbolic link, if you have write permission in the directory.

If either of the files *.* (dot) or *..* (dot, dot) are specified as the base name portion of the *File* parameter, the **rm** command writes a diagnostic message to standard error and does nothing more with such parameters.

The **rm** command writes a prompt to standard error and reads a line from standard input if the **-f** flag is not specified, and either the *File* parameter does not have write permission and the standard input is a workstation, or the **-i** flag is specified. If the response is not affirmative, the **rm** command does nothing more with the current file and proceeds to the next file.

The files owned by other users cannot be removed if the sticky bit of the directory is set and the directory is not owned by the user.

Note: The **rm** command supports the **—** (dash, dash) parameter as a delimiter that indicates the end of the flags.

An attempt to remove a file or directory that has been exported for use by the NFS version 4 server will fail with a message saying that the resource is busy. The file or directory must be unexported for NFS version 4 use before it can be removed.

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 effect.
- 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 effect.
- 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.

rmbdsp command

Purpose

Remove a backing device and return the storage back to the storage pool.

Syntax

```
rmbdsp {-sp StoragePool-bd LogicalVolume | -vtd VirtualTargetDevice} [-savebd][-f]
```

Description

The **rmbdsp** command removes a backing device from a virtual SCSI server adapter by removing its associated virtual target device. By default the backing device is also removed and its storage returned to the storage pool. If the **-savebd** flag is included then the backing device is not removed. The backing device can be identified by either specifying the name, **-bd**, or the virtual target device, **-vtd**. If the backing device is identified by the name, **-bd**, then the storage pool the device is contained within must also be specified, **-sp**. If the storage pool is not given the default storage pool is assumed.

Note:

- A storage pool must be specified if the given backup device name is a file backing device. The default storage pool is not assumed.
- The logical volume specified for the **-bd** option cannot be assigned to a shared memory pool (to be used as a paging space device by a shared memory partition).

If **-vtd** flag is given only the specified virtual target device is removed. If **-bd** flag is given all virtual target devices associated with the backing device will be removed.

Flags

-bd <i>BackingDevice</i>	Specifies the backing device that is the backing device.
-savebd	Instructs the command not to remove the backing device.
-sp	Specifies the storage pool the backing device is within.
-vtd <i>VirtualTargetDevice</i>	Specifies the virtual target devices that associate the backing device with the virtual SCSI adapter.
-f	Forces the metadata file associated to the file-backed disk to be removed. If you also specify the -savebd flag, the backing device will not be removed.

CAUTION:

Only use this flag if the metadata file associated to the file-backed disk is corrupt.

Exit Status

25 Specified logical volume is not a backing device.

Examples

To remove the virtual target device vtscsi4 and not remove the backing device associated with it, type:

```
rmbdsp -vtd vtscsi4 -savebd
```

rmdev command

Purpose

Removes a device from the system.

Syntax

```
rmdev { -dev | -pdev } Name [ -recursive ] [ [ -ucfg ]
```

Description

The **rmdev** command unconfigures and undefines the device specified with the device logical name. If you specify the **-recursive** flag, the **rmdev** command acts on any children of the device as well. By specifying the **-ucfg** flag the device will be unconfigured but not undefined.

Use the **-pdev** flag along with the parent device's logical name to delete all of the children devices. The children are deleted in the same recursive fashion as described above for the **-recursive** flag. The only difference is that the specified device itself is not deleted. Thus, the **-recursive** flag is redundant and need not be specified with the **-pdev** flag.

Attention: To protect the Configuration database, the **rmdev** command cannot be interrupted. Stopping this command before execution is complete could result in a corrupted database.

Note: The device specified or the children of the device specified cannot be a physical or logical volume that is assigned to a shared memory pool (to be used as a paging space device by a shared memory partition).

Flags

-dev <i>Name</i>	Specifies the logical device, indicated by the <i>Name</i> parameter. This flag may not be used with the -pdev flag.
-pdev <i>Name</i>	Specifies the parent logical device (indicated by the <i>Name</i> parameter whose children need to be removed. This flag may not be used with the -dev flag.
-recursive	Unconfigures the device and its children.
-ucfg	Unconfigures, but does not undefine, the specified device. The device's state will be moved from Available to Defined. To move the device back to Available state run <code>cfgdev -dev <i>Name</i></code>

Exit Status

See "Exit status for Virtual I/O Server commands" on page 1.

Examples

1. To unconfigure the **cd0** CD-ROM device, type:

```
rmdev -dev cd0
```
2. To unconfigure the SCSI adapter **scsi1** and all of its children, type:

```
rmdev -recursive -dev scsi1
```
3. To unconfigure just the children of the SCSI adapter **scsi1**, but not the adapter itself, type:

```
rmdev -pdev scsi1
```
4. To unconfigure the children of PCI bus **pci1** and all other devices under them, type:

```
rmdev -pdev pci1
```

Related Information

The `cfgdev` command, the `chdev` command, the `lsdev` command, the `mkvdev` command, and the `rmdev` command.

rmlv command

Purpose

Removes logical volumes from a volume group.

Syntax

```
rmlv [ -f ] LogicalVolume ...
```

Description

The `rmlv` command removes a logical volume. The *LogicalVolume* parameter can be a logical volume name or logical volume ID.

Attention: This command destroys all data in the specified logical volumes. The specified logical volume cannot be assigned to a shared memory pool (to be used as a paging space device by a shared memory partition).

Flags

`-f` Removes the logical volumes without requesting confirmation.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

1. 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.

Related Information

The `mklv` command, the `extendlv` command, and the `lslv` command.

rmlvcopy command

Purpose

Removes a copy of a logical volume.

Syntax

```
rmlvcopy LogicalVolume [ PhysicalVolume ... ]
```

Description

The **rmlvcopy** command removes one of the copies (disabling mirroring) of the logical volume. 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 the copy from that physical volume will be removed.

Note: If the LVM (Logical Volume Manager) has not recognized that a disk has failed, it is possible that the LVM will remove a different mirror. Therefore, if you know that a disk has failed and the LVM does not show those disks as missing, you should specify the failed disks on the command line.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

To remove mirroring from the logical volume **lv0112**, type:

```
rmlvcopy lv0112
```

Related Information

The **mklv** command, the **extendlv** command, the **rmlv** command, and the **lslv** command.

rmpath command

Purpose

Removes from the system a path to an MPIIO-capable device.

Syntax

```
rmpath { [ -dev Name ] [ -pdev Parent ] [ -conn Connection ] } [ -rm ]
```

Description

The **rmpath** command unconfigures, and possibly undefines, one or more paths associated with the specified target device (**-dev** *Name*). The set of paths that is removed is determined by the combination of the **-dev** *Name*, **-pdev** *Parent* and **-conn** *Connection* flags. If the command will result in all paths associated with the device being unconfigured or undefined, the command will exit with an error and without unconfiguring or undefining any path. In this situation, **rmdev** command must be used instead to unconfigure or undefine the target device itself.

The default action unconfigures each specified path, but does not completely remove it from the system. If the **-rm** flag is specified, the **rmpath** command unconfigures (if necessary) and removes, or deletes, the path definition(s) from the system.

When the **rmpath** command finishes, it displays a status message. When unconfiguring paths, it is possible for this command to be able to unconfigure some paths and not others (for example, paths that are in the process of doing I/O cannot be unconfigured).

The **rmpath** command provides status messages about the results of operation. Messages in one of the following formats will be generated:

```
path [ defined | deleted ]
```

This message is displayed when a single path was successfully unconfigured or undefined. If the

path is successfully configured the message path available displays. If the path is not successfully configured and there is no explicit error code returned by the method, the message path defined displays.

paths [defined | deleted]

This message is displayed if multiple paths were identified and all paths were successfully unconfigured or undefined. If the **-rm** flag is not specified, the message would be paths defined. If the **-rm** flag is specified, the message would be paths deleted.

some paths [defined | deleted]

This message is display if multiple paths were identified, but only some of them were successfully unconfigured or undefined. If the **-rm** flag is not specified, the message would be some paths defined. If the **-rm** flag is specified, the message would be some paths deleted.

no paths processed

This message is generated if no paths were found matching the selection criteria.

Flags

-rm	Indicates that the specified paths are to be deleted from the system.
-dev <i>Name</i>	Specifies the logical device name of the target device whose path is to be removed. The paths to be removed are qualified via the -pdev and -conn flags.
-pdev <i>Parent</i>	Indicates the logical device name of the parent device to use in qualifying the paths to be removed. Since all paths to a device cannot be removed by this command, either this flag, the -conn flag, or both must be specified.
-conn <i>Connection</i>	Indicates the connection information to use in qualifying the paths to be removed. Since all paths to a device cannot be removed by this command, either this flag, the -pdev flag, or both must be specified.

Exit Status

See "Exit status for Virtual I/O Server commands" on page 1.

Examples

1. To unconfigure the path from **scsi0** to **hdisk1** at connection 5,0, type:

```
rmpath -dev hdisk1 -pdev scsi0 -conn "5,0"
```

The message generated would be similar to:

```
path defined
```

2. To unconfigure all paths from **scsi0** to **hdisk1**, type:

```
rmpath -dev hdisk1 -pdev scsi0
```

If all paths were successfully unconfigured, the message generated would be similar to:

```
paths defined
```

However, if only some of the paths were successfully unconfigured, the message would be similar to:

```
some paths defined
```

3. To undefine the path definition between **scsi0** and **hdisk1** at connection 5,0, type:

```
rmpath -rm -dev hdisk1 -pdev scsi0 -conn "5,0"
```

The message generated would be similar to the following:

```
path deleted
```

4. To unconfigure all paths from **scsi0** to **hdisk1**, type:

```
rmpath -rm -dev hdisk1 -pdev scsi0
```

The message generated would be similar to:
paths deleted

Related Information

The `chpath` command, the `lspath` command, and the `rmpath` command.

rmrep command

Purpose

Remove the Virtual Media Repository

Syntax

```
rmrep [-f ]
```

Description

The `rmrep` command removes the Virtual Media Repository from the Virtual I/O Server. If there are any Virtual Target Devices associated with any virtual optical media in the repository the command will fail. If virtual optical media exist in the repository, but there are no Virtual Target Devices associated with them, the command will fail unless the `-f` flag is specified.

Flags

`-f` Forces the repository to be removed even if it contains virtual optical media.

Examples

To remove the Virtual Media Repository, type the following command:

```
rmrep
```

rmfsp command

Purpose

Removes a file storage pool.

Syntax

```
rmfsp [ -f StoragePool ]
```

Description

The `rmfsp` command removes the specified storage pool from the Virtual I/O Server. This command only works for file storage pools. To remove a logical volume storage pool, use the `chfsp` command to remove all physical volumes from the pool. If the pool contains any files, the `-f` flag must be specified or the command will fail. Also, any virtual target devices associated with files in the pool must be removed prior to running the `rmfsp` command.

Flags

`-f` Forces the pool to be removed even if it contains files.

Example

To remove the storage pool `Client_Data`, type the following command:

```
rmsp Client_Data
```

IVM `rmsyscfg` command

Purpose

Removes a logical partition from the managed system. This command is valid only in an Integrated Virtualization Manager environment.

Syntax

```
rmsyscfg -r lpar { -n PartitionName | --id PartitionID }
```

Description

The `rmsyscfg` command removes a logical partition from the managed system.

Flags

-r <i>ResourceType</i>	The type of resources to remove:
-m <i>ManagedSystem</i>	lpar - Logical partition resources The name of the managed system. This attribute is optional because there is only one system to manage. The name may either be the user-defined name for the managed system, or be in the form <code>ttt-mmm*sssssss</code> , where <code>ttt</code> is the machine type, <code>mmm</code> is the model, and <code>sssssss</code> is the serial number of the managed system.
-n <i>PartitionName</i>	The name of the partition which you want to remove.
--id <i>PartitionID</i>	The partition ID of the partition to remove.

Exit Status

This command has a return code of zero on success.

Security

This command is not accessible by users with the `ViewOnly` role.

Examples

1. To delete a partition with an `lpar` ID of 3, type:

```
rmsyscfg -r lpar --id 3
```
2. To delete a partition a name of `lp3`, type:

```
rmsyscfg -r lpar -n lp3
```

Related Information

The `Issyscfg` command, the `mksyscfg` command, and the `chsyscfg` command.

IVM rmsysplan command

Purpose

Removes a system plan file.

Syntax

```
rmsysplan [-f <system plan file name> [--help]
```

Description

The `rmsysplan` command removes a system plan file.

Flags

<code>-f [SysPlanFileName]</code>	Specifies the system plan file name that will be removed from the Integrated Virtualization Manager system plan file directory. File name should have suffix <code>.sysplan</code>
<code>--help</code>	Displays the help text for this command and then exits.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

To remove a system plan file `test.sysplan`, type the following command:

```
rmsysplan -f test.sysplan
```

Related Information

The `deploysysplan` command, `mksysplan` command, and the `lssysplan` command.

IVM rmsysplanres command

Purpose

Remove the system plan resource.

Syntax

```
rmsysplanres -r osinstall -n <resource name> [-help]
```

Description

The `rmsysplanres` command removes a system plan resource that is defined on the Integrated Virtualization Manager (IVM).

Flags

<code>-r</code>	Specifies the type of system plan resources to remove. To remove the installation resources for the operating system, the only valid value is <code>osinstall</code> .
<code>-n</code>	Specifies the name of the system plan resource to remove.
<code>-help</code>	Displays the help text for this command and exit.

Exit Status

The following exit values are returned:

0	Successful completion.
>0	Invalid flag, argument, or command failure.

Examples

1. To remove the system plan resource for the **osinstall** type used as the Red Hat installation source, type:

```
rmsysplanres -r osinstall -n rhel4
```
2. To remove the system plan resource for the **osinstall** type used as the AIX 5.3 installation source, type:

```
rmsysplanres -r osinstall -n aix53
```

Related Information

The `defsysplanres`, `deploysysplan`, `lssysplanres`, and `lssysplan` commands.

rmtcpip command

Purpose

Removes the Virtual I/O Server TCP/IP configuration.

Syntax

```
rmtcpip [-f] [-nextboot] {-all | [-hostname] [-routing] [-interface ifnameList]}
```

```
rmtcpip [-f] {-all | [-namesrv] [-hostname] [-routing] [-interface ifnameList]}
```

Description

The **rmtcpip** command removes Virtual I/O Server TCP/IP settings, such as name server information, network interface configuration, routing information, and host name.

Flags

-all	Removes all TCP/IP settings, effectively resetting it to a newly installed system.
-f	Executes the command with out prompting for user confirmation.
-interface	Removes TCP/IP configuration from listed interfaces.
-hostname	Resets the hostname to <code>ioserver</code> .
-namesrv	Removes DNS information and clears the hosts file.
-nextboot	Removes the specified information from the configuration files, leaving the current network parameters intact (all except DNS information and hosts file).
-routing	Removes static routing tables.

Examples

1. To remove all Virtual I/O Server TCP/IP configuration, type:

```
rmtcpip -all
```

Answer yes when prompted

2. To unconfigure a network interface en0 without confirmation, type:
`rmtcpip -f -interface en0`
3. To cleanup the static routing table, type:
`rmtcpip -f -routing`
4. To remove IP information from a network interface on the next boot, keeping the current configuration running execute:
`rmtcpip -f -interface en0 -nextboot`

rmuser command

Purpose

Removes a user account.

Syntax

```
rmuser [-ldap] [-rmdir] Name
```

Description

The **rmuser** command removes the user account that is identified by the *Name* parameter. This command removes a user account attribute. If you specify the **rmdir** flag, this command also removes the user's home directory and files.

Flags

- | | |
|---------------|---|
| -rmdir | Removes the specified user's home directory and files. |
| -ldap | Attention: This will delete all data stored in this user account's home directory.
Identifies the user as an LDAP user account. |

Exit Status

See "Exit status for Virtual I/O Server commands" on page 1.

Security

This command can only be run by the prime administrator (padmin) user.

Examples

1. To remove user account **haarlem**, type the following command:
`rmuser haarlem`
2. To remove the user account and home directory of user account **emmen**, type the following command:
`rmuser -rmdir emmen`

Attention: This will delete all data stored in this user account's home directory.

Related Information

The chuser command, the lsuser command, the mkuser command, and the passwd command.

rmvdev command

Purpose

To remove the connection between a physical device and its associated virtual SCSI adapter.

Syntax

```
rmvdev [ -f ] { -vdev TargetDevice | -vtd VirtualTargetDevice } [-rmlv]
```

Description

The **rmvdev** command removes the connection between a physical device and its associated virtual SCSI adapter. The connection can be identified by specifying the backing (physical) device or the virtual target device. If the connection is specified by the device name and there are multiple connections between the physical device and virtual SCSI adapters and error is returned unless the **-f** flag is also specified. If **-f** is included then all connections associated with the physical device are removed.

If the backing (physical) device is a logical volume and the **-rmlv** flag is specified, then logical volume will be removed as well.

Flags

-f	Forces the removal of all virtual target devices associated with the give backing device.
-vdev <i>TargetDevice</i>	Specifies the physical backing device
-rmlv	Deletes the backing device. This flag is valid only for logical volume backing devices.
-vtd <i>VirtualTargetDevice</i>	Specifies the virtual target device to remove.

rmvopt command

Purpose

Remove a virtual optical media disk from the Virtual Media Repository.

Syntax

```
rmvopt [-f ] -name FileName
```

Description

The **rmrep** command removes the specified media from the Virtual Media Repository. If the media is currently loaded into one or more virtual optical devices the command will fail, unless the **-f** flag is specified.

Flags

-f	Force the media to be removed even if it is loaded into one or more virtual optical devices.
-name <i>FileName</i>	The name of the virtual optical media to remove.

Examples

To remove the file `clientData` from the Virtual Media Repository, type the following command:

```
rmvopt -name clientData
```

rmvt command

Purpose

Closes a virtual terminal connection to a partition. This command is valid only in an Integrated Virtualization Manager environment.

Syntax

```
rmvt { -id lparID }
```

Description

The **rmvt** command closes a virtual terminal connection to the target logical partition. To close the virtual terminal session normally, enter the ~. character sequence in the terminal window.

This command requires additional HMC configuration if used in an HMC environment.

Flags

-id *lparID* The ID of the partition for which to close the virtual terminal session.

Exit Status

29 Virtual terminal device is not found

Security

This command is not accessible by users with the ViewOnly role.

Examples

1. To close a virtual terminal connection to the partition with ID 3, type:

```
rmvt -id 3
```

Related Information

The **mkvt** command.

IVM rsthwres command

Purpose

Restores hardware resources

Syntax

```
rsthwres -r io | mem | proc | virtualio | mempool [-m <managed system>] [-p <partition name> | --id <partition ID>] [-l <DRC index>] [-s <virtual slot number>] [-a "<attributes>"] [--help]
```

```
rsthwres -rhea | -m <managed system>] [-p <partition name> | --id <partition ID>] [-l<HEA DRC index>] [-g <port group ID>] --logport <logical port ID>]--help]
```

To restore memory or processing resources

rsthwres -r {*mem* | *proc*} [{-**p** *partition-name* | --**id** *partition-ID*}]

To restore physical I/O slots

rsthwres -r io [{-**p** *partition-name* | --**id** *partition-ID*}] [-**I** *slot-DRC-index*]

To restore Host Ethernet Adapter resources

rsthwres -r hea [{-**p** *partition-name* | --**id** *partition-ID*}] [-**IHEA**-*adapter-ID*] [-**g** *port-group* --**logport** *logical-port-ID*]

To restore memory pool resources

rsthwres -r mempool [-**a** "<attributes>"] [--**help**]

Description

The **rsthwres** command restores the hardware resource configuration in the managed system. This operation may need to be performed after a dynamic LPAR operation fails.

You also need to perform this operation to restore memory pool resources in the following situations:

- When you reinstall the VIOS base code and applied fix packs from original VIOS media.
- When you restore the VIOS from backup media prior to the time you created the memory pool, but the firmware still has a memory pool configured.

Flags

-r	Shows the type of hardware resources to restore: io I/O slot (physical) hea Host Ethernet Adapter mem Memory mempool Memory pools proc Processing virtualio Virtual I/O
-m < <i>managed system</i> >	Indicates the name of the managed system which has the partitions for which to restore the hardware resources. The name may either be the user-defined name for the managed system, or be in the form <i>ttt-mmm*sssssss</i> , where <i>ttt</i> is the machine type, <i>mmm</i> is the model, and <i>sssssss</i> is the serial number of the managed system.
-p < <i>partition name</i> >	Indicates the name of the partition for which to restore the hardware resources. To restore hardware resources for a single partition, you must either use this option to specify the name of the partition, or use the -- id option to specify the partition's ID. Otherwise, hardware resources for all partitions in the managed-system will be restored. The -p and the -- id options are mutually exclusive.

--id <partition ID>	Indicates the ID of the partition for which to restore the hardware resources. To restore hardware resources for a single partition, you must either use this option to specify the ID of the partition, or use the -p option to specify the partition's name. Otherwise, hardware resources for all partitions in the managed-system will be restored.
-l <DRC index>	The --id and the -p options are mutually exclusive. Indicates the DRC index of the physical I/O slot to restore. This option is only valid when restoring physical I/O slots.
-s <virtual slot>	This option is also used to specify the physical Host Ethernet Adapter to restore. Indicates the slot number of the virtual I/O.
-a "<attributes>"	The configuration data needed to set hardware resource related attributes. The configuration data consists of attribute name/value pairs, which are in comma separated value (CSV) format. The configuration data must be enclosed in quotation marks. Possible values are: paging_storage_pool Note: <ul style="list-style-type: none"> • If you intend to use a paging storage pool, you must specify the value here. After you run the <code>rsthwres</code> command, you cannot set or change the paging storage pool value without deleting your memory pool and recreating it. • The -a parameter is valid only with -r mempool, and the only supported attribute is <code>paging_storage_pool</code>.
-g port group ID	Indicates the logical ports that are part of the specified port group that will be restored.
--logport <virtual slot>	Indicates the ID of the logical port to recover.
--help	Displays the help text for this command and exit.

Exit Status

This command has a return code of 0 on success.

Examples

To restore the physical I/O slots for all partitions type the following command:

```
rsthwres -r io -m mySystem
```

To restore the physical I/O slot with DRC index **21010003**, type the following command:

```
rsthwres -r io -m 9406-570*12345678 -l 21010003
```

To restore memory resources for partition **p1**, type the following command:

```
rsthwres -r mem -m 9406-570*12345678 -p p1
```

To restore processing resources for the partition with ID **1**, type the following command:

```
rsthwres -r proc -m mySystem --id 1
```

To clean up all recoverable Host Ethernet Adapter resources, type the following command:

```
rsthwres -r hea
```

To clean up a specific logical port with a given partition, type the following command:

```
rsthwres -r hea -l <HEA DRC> -g <PORT GROUP>  
--logport <LP ID> {-p <LPAR NAME | --id <LPAR ID>}
```

To recover a memory pool using rootvg as the paging storage pool, type the following command:

```
rsthwres -r mempool -a paging_storage_pool=rootvg
```

Note: If you intend to use a paging storage pool, you must specify the value here. After you run the rsthwres command, you cannot set or change the paging storage pool value without deleting your memory pool and recreating it.

IVM rstprofdata command

Purpose

Restores profile data. This command is valid only in an Integrated Virtualization Manager environment.

Syntax

To restore logical partition configuration data from a file:

```
rstprofdata -l RestoreType [-f RestoreFile] [ --ignoremtms ] [ --ignoremac ] [ --ignorehea ] [ -m ManagedSystem ]
```

Description

The **rstprofdata** command performs a restore of logical partition configuration information from a file which was created with the **bkprofdata** command. Logical partition configuration must be cleared using the **lpcfgop** command before performing a restore operation. The restore operation may require a restart of the managed system, after which the **rstprofdata** command should be run again with the same parameters as were used the first time (before the system restart).

Flags

-l <i>RestoreType</i>	The type of restore to perform. Valid options are: 1 - full restore from the backup file
-f <i>RestoreFile</i>	The name of the file to read from in the current working directory. If not specified, the default file is <code>"/var/adm/lpm/profile.bak"</code>
--ignoremtms	Do not fail the restore operation if the type, model, or serial number of the managed system does not match the values in the backup file. This flag is used for the following primary reasons: <ul style="list-style-type: none">• To clone the configuration from one system to another.• To force new worldwide port names to be allocated for any logical partitions that use virtual fibre channel. Note: If you use the -ignoremtms flag, new worldwide port names are generated.
--ignoremac	Do not try to restore the virtual Ethernet MAC addresses from the backup file. Default to the automatically generated MAC addresses.
--ignorehea	Do not try to restore the Host Ethernet Adapter resources.
-m <i>ManagedSystem</i>	The name of the managed system. This attribute is optional because there is only one system to manage. The name may either be the user-defined name for the managed system, or be in the form <code>ttt-mmm*sssssss</code> , where <code>ttt</code> is the machine type, <code>mmm</code> is the model, and <code>sssssss</code> is the serial number of the managed system.
--help	Display the help text for this command and exit.

Exit Status

This command has a return code of 0 on success.

Security

This command is not accessible by users with the ViewOnly role.

Examples

1. To restore the partition configuration data from `/var/adm/lpm/profile.bak`, type:

```
rstprofdata -l 1
```
2. To restore the partition configuration data from `lparData.bak` without validating that the type, model, and serial number match, type:

```
rstprofdata -l 1 -f lparData.bak --ignoremtms
```

Related Information

The `bkprofdata` command.

save_base command

Purpose

Saves information about base-customized devices in the Device Configuration database for the boot device.

Syntax

```
save_base [-path Directory][-file File][-verbose]
```

Description

The `save_base` command stores customized information for base devices to use during phase 1 of system boot. By default, the `save_base` command retrieves this information from the `/etc/objrepos` directory. However, you can override this action by using the `-o` flag to specify an ODM directory. The `save_base` command is typically run without any parameters. It uses the `/dev/ipl_blv` special file link to identify the output destination.

Alternatively, use the `-d` flag to specify a destination file or a device, such as the `/dev/hdisk0` device file. To identify a specific output destination, the `-d` flag identifies the file to which `save_base` writes the base customized device data. This file can be either a regular file or a device special file. The device special file identifies either a disk device special file or a boot logical volume device special file.

A disk device special file can be used where there is only one boot logical volume on the disk. The `save_base` command ensures that the given disk has only one boot logical volume present. If neither of these conditions is true, `save_base` does not save the base customized device data to the disk and exits with an error.

When a second boot logical volume is on a disk, the boot logical volume device special file must be used as the destination device to identify which boot image the base customized device data will be stored in. A boot logical volume device special file can be used even if there is only one boot logical volume on the disk. The `save_base` command ensures that the given device special file is a boot logical volume and it is bootable before saving any data to it. If either of these checks fails, `save_base` exits with an error.

The `save_base` command determines what device information to save using the `PdDv.base` field corresponding to each entry in the `CuDv` object class. Specifically, the `PdDv.base` field is a bit mask which represents the type of boot for which this device is a base device. The `save_base` command determines the current type of boot by accessing the `boot_mask` attribute in the `CuAt` object class. The value of this attribute is the bit mask to apply to the `PdDv.base` field to determine which devices are base.

Note:

- Base devices are those devices that get configured during phase 1 boot; they may vary depending on the type of boot (mask). For example, if the mask is NETWORK_BOOT, network devices are considered base; for DISK_BOOT, disk devices are considered base. The type of boot masks are defined in the /usr/include/sys/cfgdb.h file.
- The **-m** flag is no longer used by the **save_base** command. For compatibility reasons, the flag can be specified, but is not used.

Flags

-path <i>Directory</i>	Specifies a directory containing the Device Configuration database. Note: By default, the save_base command retrieves this information from the /etc/objrepos directory.
-file <i>file</i>	Specifies the destination file or device to which the base information will be written.
-verbose	Specifies that verbose output is to be written to standard output (STDIN).

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

1. To save the base customized information and see verbose output, type the following command:
`save_base -verbose`
2. To specify an ODM directory other than the /usr/lib/objrepos directory, type the following command:
`save_base -o /tmp/objrepos`
3. To save the base customized information to the /dev/hdisk0 device file instead of to the boot disk, type the following command:
`save_base -d /dev/hdisk0`

savevgstruct command**Purpose**

Backs up a volume group.

Syntax

```
savevgstruct VolumeGroupLabel
```

Description

The **savevgstruct** command will make a backup of a volume group structure.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

1. To save the structure of the user defined volume group **myvg**, enter:
`savevgstruct myvg`

Related Information

The activatevg command, the restorevgstruct command, the chvg command, the deactivatevg command, the exportvg command, the extendvg command, the importvg command, the lsvg command, the mkvg command, and the syncvg command.

seastat command

Purpose

Generates a report to view, per client, Shared Ethernet Adapter statistics.

Syntax

To display Shared Ethernet Adapter statistics, per client.

```
seastat -d Shared Ethernet Adapter device name [-n]
```

To clear all Shared Ethernet Adapter statistics that have been gathered per client.

```
seastat -d Shared Ethernet Adapter device name -c
```

Description

The seastat command generates a report to view, per client, shared ethernet adapter statistics. To gather network statistics at a per-client level, advanced accounting can be enabled on the Shared Ethernet Adapter to provide more information about its network traffic. To enable per-client statistics, the VIOS administrator can set the Shared Ethernet Adapter accounting attribute to enabled. The default value is disabled. When advanced accounting is enabled, the Shared Ethernet Adapter keeps track of the hardware (MAC) addresses of all of the packets it receives from the LPAR clients, and increments packet and byte counts for each client independently. After advanced accounting is enabled on the Shared Ethernet Adapter, the VIOS administrator can generate a report to view per-client statistics by running the **seastat** command.

Note: Advanced accounting must be enabled on the Shared Ethernet Adapter before the **seastat** command can print any statistics.

To enable advanced accounting on the SEA, enter the following command:

```
chdev -dev <SEA device name> -attr accounting=enabled
```

Flags

-d <i>Shared Ethernet Adapter device name</i>	Specifies the device name of the Shared Ethernet Adapter.
-n	Disables name resolution on the IP addresses.
-c	Clears all of the per client Shared Ethernet Adapter statistics that have been gathered.

Exit Status

The following exit values are returned:

0	Successful completion.
>0	Invalid flag, argument, or command failure

Examples

1. To display Shared Ethernet Adapter statistics for **sea0**, type:
`seastat -d sea0`
2. To clear the Shared Ethernet Adapter statistics for **sea0**, type:
`seastat -d sea0 -c`

sed command

Purpose

A stream editor.

Syntax

```
sed [ -n ] Script [ File ... ]
```

```
sed [ -n ] [ -e Script ] ... [ -f ScriptFile ] ... [ File ... ]
```

Description

The **sed** command modifies lines from the specified *File* parameter according to an edit script and writes them to standard output. The **sed** command includes many features for selecting lines to be modified and making changes only to the selected lines.

The **sed** command uses two work spaces for holding the line being modified: the pattern space, where the selected line is held; and the hold space, where a line can be stored temporarily.

An edit script consists of individual subcommands, each one on a separate line. The general form of **sed** subcommands is the following:

```
[address-range] function[modifiers]
```

The **sed** command processes each input *File* parameter by reading an input line into a pattern space, applying all **sed** subcommands in sequence whose addresses select that line, and writing the pattern space to standard output. It then clears the pattern space and repeats this process for each line specified in the input *File* parameter. Some of the **sed** subcommands use a hold space to save all or part of the pattern space for subsequent retrieval.

When a command includes an address (either a line number or a search pattern), only the addressed line or lines are affected by the command. Otherwise, the command is applied to all lines.

An address is either a decimal line number, a \$ (dollar sign), which addresses the last line of input, or a context address. A context address is a regular expression similar to those used in the **ed** command except for the following differences:

- You can select the character delimiter for patterns. The general form of the expression is:
`\?pattern?`

where ? (question mark) is a selectable character delimiter. You can select any character from the current locale except for the space or new-line character. The \ (backslash) character is required only for the first occurrence of the ? (question mark).

The default form for the pattern is the following:

```
/pattern/
```

A \ (backslash) character is not necessary.

- The `\n` sequence matches a new-line character in the pattern space, except the terminating new-line character.
- A `.` (period) matches any character except a terminating new-line character. That is, unlike the `ed` command, which cannot match a new-line character in the middle of a line, the `sed` command can match a new-line character in the pattern space.

Certain commands called *addressed* commands allow you to specify one line or a range of lines to which the command should be applied. The following rules apply to addressed commands:

- A command line without an address selects every line.
- A command line with one address, expressed in context form, selects each line that matches the address.
- A command line with two addresses separated by commas selects the entire range from the first line that matches the first address through the next line that matches the second. (If the second address is a number less than or equal to the line number first selected, only one line is selected.) Thereafter, the process is repeated, looking again for the first address.

Flags

<code>-e Script</code>	Uses the <i>Script</i> variable as the editing script. If you are using just one <code>-e</code> flag and no <code>-f</code> flag, the <code>-e</code> flag can be omitted.
<code>-f ScriptFile</code>	Uses the <i>ScriptFile</i> variable as the source of the edit script. The <i>ScriptFile</i> variable is a prepared set of editing commands applied to the <i>File</i> parameter.
<code>-n</code>	Suppresses all information normally written to standard output.

Note: You can specify multiple `-e` and `-f` flags. All subcommands are added to the script in the order specified, regardless of their origin.

sed Subcommands

The `sed` command contains the following `sed` script subcommands. The number in parentheses preceding a subcommand indicates the maximum number of permissible addresses for the subcommand.

Note:

1. The *Text* variable accompanying the `a\`, `c\`, and `i\` subcommands can continue onto more than one line, provided all lines but the last end with a `\` (backslash) to quote the new-line character. Backslashes in text are treated like backslashes in the replacement string of an `s` command and can be used to protect initial blanks and tabs against the stripping that is done on every script line. The *RFile* and *WFile* variables must end the command line and must be preceded by exactly one blank. Each *WFile* variable is created before processing begins.
2. The `sed` command can process up to 999 subcommands in a pattern file.

Subcommand	Description
(1) <code>a\</code> <i>Text</i>	Places the <i>Text</i> variable in output before reading the next input line.
(2) <code>b[label]</code>	Branches to the <code>:</code> command bearing the <i>label</i> variable. If the <i>label</i> variable is empty, it branches to the end of the script.
(2) <code>c\</code> <i>Text</i>	Deletes the pattern space. With 0 or 1 address or at the end of a 2-address range, places the <i>Text</i> variable in output and then starts the next cycle.
(2) <code>d</code>	Deletes the pattern space and then starts the next cycle.
(2) <code>D</code>	Deletes the initial segment of the pattern space through the first new-line character and then starts the next cycle.
(2) <code>g</code>	Replaces the contents of the pattern space with the contents of the hold space.
(2) <code>G</code>	Appends the contents of the hold space to the pattern space.

Subcommand	Description
(2) h	Replaces the contents of the hold space with the contents of the pattern space.
(2) H	Appends the contents of the pattern space to the hold space.
(1) i <i>Text</i>	Writes the <i>Text</i> variable to standard output before reading the next line into the pattern space.
(2) l	Writes the pattern space to standard output showing nondisplayable characters as 4-digit hexadecimal values. Long lines are folded.
(2) l	Writes the pattern space to standard output in a visually unambiguous form. The characters <code>\\</code> , <code>\\a</code> , <code>\\b</code> , <code>\\f</code> , <code>\\r</code> , <code>\\t</code> , and <code>\\v</code> are written as the corresponding escape sequence. Non-printable characters are written as 1 three-digit octal number (with a preceding backslash character) for each byte in the character (most significant byte first). This format is also used for multibyte characters. This subcommand folds long lines. A backslash followed by a new-line character indicates the point of folding. Folding occurs at the 72nd column position. A \$ (dollar sign) marks the end of each line.
(2) n	Writes the pattern space to standard output if the default output is not suppressed. It replaces the pattern space with the next line of input.
(2) N	Appends the next line of input to the pattern space with an embedded new-line character (the current line number changes). You can use this to search for patterns that are split onto two lines.
(2) p	Writes the pattern space to standard output.
(2) P	Writes the initial segment of the pattern space through the first new-line character to standard output.
(1) q	Branches to the end of the script. It does not start a new cycle.
(2) r <i>RFile</i>	Reads the contents of the <i>RFile</i> variable. It places contents in output before reading the next input line.
(2) s / <i>pattern</i> / <i>replacement</i> / <i>flags</i>	Substitutes the <i>replacement</i> string for the first occurrence of the <i>pattern</i> parameter in the pattern space. Any character that is displayed after the s subcommand can substitute for the <i>/</i> (slash) separator except for the space or new-line character. The value of the <i>flags</i> variable must be zero or more of: g Substitutes all non-overlapping instances of the <i>pattern</i> parameter rather than just the first one. n Substitutes for the <i>n</i> -th occurrence only of the <i>pattern</i> parameter. p Writes the pattern space to standard output if a replacement was made. w <i>WFile</i> Writes the pattern space to the <i>WFile</i> variable if a replacement was made. Appends the pattern space to the <i>WFile</i> variable. If the <i>WFile</i> variable was not already created by a previous write by this sed script, the sed command creates it.
(2) t <i>label</i>	Branches to the <i>:label</i> variable in the script file if any substitutions were made since the most recent reading of an input line execution of a t subcommand. If you do not specify the <i>label</i> variable, control transfers to the end of the script.
(2) w <i>WFile</i>	Appends the pattern space to the <i>WFile</i> variable.
(2) x	Exchanges the contents of the pattern space and the hold space.
(2) y / <i>pattern1</i> / <i>pattern2</i>	Replaces all occurrences of characters in the <i>pattern1</i> variable with the corresponding <i>pattern2</i> characters. The number of characters in the <i>pattern1</i> and <i>pattern2</i> variables must be equal. The new-line character is represented by <code>\n</code> .
(2) ! <i>sed-cmd</i>	Applies the specified sed subcommand only to lines not selected by the address or addresses.
(0): <i>label</i>	Marks a branch point to be referenced by the b and t subcommands. This label can be any sequence of eight or fewer bytes.
(1)=	Writes the current line number to standard output as a line.
(2){ <i>subcmd</i> }	Groups subcommands enclosed in {} (braces).
(0)	Ignores an empty command.

Subcommand

(0)#

Description

If a # (pound sign) appears as the first character on a line of a script file, that entire line is treated as a comment, with one exception. For the first line of a script file only, if the character after the # is an n, the default output will be suppressed. The rest of the line after the #n is ignored.

Exit Status

This command returns the following exit values:

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

Examples

1. To perform a global change, enter:

```
sed "s/happy/enchanted/g" chap1
```

This command sequence replaces each occurrence of the word `happy` found in the file `chap1` with the word `enchanted`. The `g` character at the end of the `s` subcommand tells the `sed` command to make as many substitutions as possible on each line. Without the `g` character, the `sed` command replaces only the first occurrence of the word `happy` on a line.

The `sed` command operates as a filter. It reads text from standard input or from the files named on the command line (`chap1` in this example), modifies this text, and writes it to standard output. Unlike most editors, it does not replace the original file. This makes the `sed` command a powerful command when used in pipelines.

2. To use the `sed` command as a filter in a pipeline, enter:

```
pr chap2 | sed "s/Page *[0-9]*$/(&)/" | enq
```

This command sequence encloses the page numbers in parentheses before printing the file `chap2`. The `pr` command puts a heading and page number at the top of each page, then the `sed` command puts the page numbers in parentheses, and the `enq` command prints the edited listing.

The `sed` command pattern `/Page *[0-9]*$/` matches page numbers that appear at the end of a line. The `s` subcommand changes this to `(&)`, where the `&` stands for the page number that was matched.

3. To display selected lines of a file, enter:

```
sed -n "/food/p" chap3
```

The `sed -n` displays each line in the file `chap3` that contains the word `food`. Normally, the `sed` command copies every line to standard output after it is edited. The `-n` flag stops the `sed` command from doing this. You then use subcommands like `p` to write specific parts of the text. Without the `-n` flag, this example displays all the lines in the file `chap3`, and it shows each line containing `food` twice.

4. To perform complex editing, enter:

```
sed -f script.sed chap4
```

This command sequence creates a `sed` script file when you want to do anything complex. You can then test and modify your script before using it. You can also reuse your script to edit other files. Create the script file with an interactive text editor.

5. A sample `sed` script file:

```
:join
/\${N}
s/\n//
b join
}
```


Examples

1. To display all exported directories on the host **middelburg**, type:
`showmount middelburg`

Related Information

The mount command and the unmount command.

shutdown command

Purpose

Ends system operation.

Syntax

```
shutdown [-force ] [ -restart ]
```

Description

The **shutdown** command halts the operating system. When the shut down is complete, you receive a shutdown completion message.

Attention: Do not attempt to restart the system or turn off the system before the shutdown completion message is displayed; otherwise, file system damage can result.

The **-force** flag is used to bypass the following user prompt:

```
"Shutting down the VIOS could affect Client Partitions. Continue [y|n]?"
```

Flags

-force	Begins a system shut down without a user prompt.
-restart	Restarts the system after being shut down.

Exit Status

See "Exit status for Virtual I/O Server commands" on page 1.

Security

This command can only be run by the prime administrator (padmin).

Examples

1. To shut down the system, type the following command:
`shutdown`
2. To restart the system, type the following command:
`shutdown -restart`
3. To shut down the system and restart, without the user prompt, type the following command:
`shutdown -force -restart`

snap command

Purpose

Gathers system configuration information.

Syntax

```
snap [-general] [-dev DeviceName]
```

```
snap script1 "script2 arg1 arg2" ...
```

Description

The **snap** command gathers system configuration information and compresses the information into a pax file (snap.pax.Z). The file can then be transmitted to a remote system. The information gathered with the **snap** command may be required to identify and resolve system problems.

At least 8 MB of temporary disk space is required to collect all system information, including contents of the error log.

Flags

-dev <i>OutputDevice</i>	Copies the compressed image onto the specified device.
-general	Gathers general system information which is a subset of the system configuration information

Exit Status

See "Exit status for Virtual I/O Server commands" on page 1.

Examples

1. To gather all system configuration information, enter the following command:

```
snap
```

The output of this command is written to the users home directory.

2. To gather general system configuration information, including the output of the **lspp -hBc** command, enter the following command :

```
snap -general -dev /dev/rfd0
```

Output is written to the **/tmp/ibmsupt/general/lspp.hBc** and **/tmp/ibmsupt/general/general.snap** files. The final snap output is copied to **/home/<User>/snap.pax.Z**. This command also writes the system information to a removable diskette.

3. To run the scripts foo1, foo2 and foo3. where foo1 takes no argument, foo2 takes three arguments and foo3 takes one argument, type the following"

```
snap foo1 "foo2 -x -y 3" "foo3 6" foo4
```

Output is written to **/tmp/ibmsupt/snapscripts/foo1**, **/tmp/ibmsupt/snapscripts/foo2** and **/tmp/ibmsupt/snapscripts/foo3** assuming the destination directory is the default, **/tmp/ibmsupt**.

Files

snap.pax.Z

snmp_info command

Purpose

Requests or modifies values of Management Information Base (MIB) variables managed by a Simple Network Management Protocol (SNMP) agent.

Syntax

```
snmp_info [-mode get | next] [-verbose] [-com Community] [-debug Level] [-host HostName] [-file ObjectFile] -retry Tries [-waitWaitTime ] [Variable Instance]
```

The following syntax is for the set option:

```
snmp_info -mode set [-verbose] [-com Community] [-debug Level] [-host HostName] [-file ObjectFile] -retry Tries [-waitWaitTime ] Variable Instance=Value
```

The following syntax is for the dump option:

```
snmp_info -mode dump[-verbose] [-com Community] [-debug Level] [-host HostName] [-file ObjectFile] -retry Tries [-waitWaitTime ] [Variable Instance]
```

Description

The **snmp_info** command requests or modifies values for one or more Management Information Base (MIB) variables managed by a Simple Network Management Protocol (SNMP) agent. This command may only be issued by a user with root privileges or by a member of the system group.

If you specify the get option, the **snmp_info** command requests information about one or more MIB variables from an SNMP agent.

If you specify the next option, the **snmp_info** command requests information from an SNMP agent about the instances following the specified instances. The next option makes it possible to obtain MIB values without knowledge of the instance qualifiers.

If you specify the set option, the **snmp_info** command modifies values for one or more MIB variables for an SNMP agent. Only a few MIB variables are designated read-write. The agent that manages the MIB database may take various actions as a side effect of modifying MIB variables. For example, setting the ifAdminStatus MIB variable to 2 will normally shut down a network interface. The action taken is determined by the implementation of the SNMP agent that manages the database.

If you specify the dump option, the **snmp_info** command can be used to traverse the entire MIB tree of a given agent. If a group is passed in as the Variable parameter, the **snmp_info** command will traverse that specified path of the MIB tree.

The **snmp_info** command has a debug facility that will dump debug information for transmitted and received packets. The facility is enabled with the **-d** flag.

Flags

-com <i>Community</i>	Specifies the community name to be used to query. If you do not specify this flag, the default community name is public.
------------------------------	--

-debug <i>Level</i>	Specifies the level of I/O debug information. Use one of the following values: 0 No debug information. 1 Port bindings and the number of bytes transmitted and received. 2 Level 1 plus a hexadecimal dump of incoming and outgoing packets. 3 Level 2 plus an English version of the request and response packets. If you do not specify this flag, the default debug level is 0.
-host <i>HostName</i>	Specifies the host name of the SNMP agent to be queried. If you do not specify this flag, the default host name is the host name of the system on which the user is currently logged.
-file <i>ObjectFile</i>	Specifies the name of the objects definition file. If you do not specify this flag, the default objects definition file name is <code>/etc/mib.defs</code> .
-mode <i>Option</i>	Specifies the mode by which to access the MIB variables. The value can be one of the following options: get Requests information about the specified MIB variables next Requests the instances following the specified instances set Modifies the specified write access MIB variables dump Dumps the specified section of the MIB tree Note: 1. The option name can be specified by the minimum number of characters required to make it unique. 2. If you do not specify this flag, the default mode is <code>get</code> .
-retry <i>Tries</i>	Specifies the number of times the <code>snmp_info</code> command transmits the SNMP request to the SNMP agent before ending with the message no SNMP response. If you do not specify this flag, the default number of tries is 3.
-verbose	Specifies that the output from the <code>snmp_info</code> command be displayed in verbose mode. If you do not specify this flag, the information will not be displayed in verbose mode.
-wait	Specifies the wait time in seconds for the response from the <code>snmpd</code> agent. If you do not specify this flag, the default wait time is 15 seconds.

Parameters

Value	Specifies the value to which the MIB Variable parameter is to be set. A value must be specified for each variable. If a value is not specified, the request packet will be invalid.
Variable	Specifies the name in text format or numeric format of a specific MIB variable as defined in the <code>/etc/mib.defs</code> file. If the option to the <code>-m</code> flag is <code>next</code> or <code>dump</code> , the Variable parameter may be specified as a MIB group.
Instance	Specifies the instance qualifier for the MIB Variable parameter. The Instance parameter is required if the option to the <code>-m</code> flag is <code>get</code> or <code>set</code> . The Instance parameter is optional if the option to the <code>-m</code> flag is <code>next</code> or <code>dump</code> . Note: 1. There should be no blank spaces in the <code>Variable.Instance</code> parameter sequence. 2. If the Instance parameter is not specified, do not place a period after the Variable parameter.

For further information, consult RFC 1213, which defines the Management Information Base (MIB) for network management, and RFC 1157, which defines the SNMP protocol for creating requests for MIB information and formatting responses.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

To display the current system information and snmp configuration information, type the following command:

```
snmp_info -mode dump system
```

Related Information

The **snmp_trap** command and the **cl_snmp** command.

snmp_trap command

Purpose

The **snmp_trap** command generates a notification (trap) to report an event to the SNMP manager with the specified message.

Syntax

```
snmp_trap [-debug] [-host HostName] [-com Community] [-target TargetHost] [-msg message]
```

Description

The **snmp_trap** command generates a notification (trap) to report an event to the SNMP manager with the specified message.

Flags

-host <i>HostName</i>	Specifies to connect to the SNMP agent on the specified host. If you do not specify this flag, the default host is the local host. Host can be an IP address or a host name.
-com <i>Community</i>	Specifies the community name to use. This community must have been set in the <code>/etc/snmpdv3.conf</code> file for SNMP version 3 file. You also have to have read-access privilege to at least the SNMP agent running on the specified host or local host. If you do not specify this flag, the default community name is "public".
-debug	Enables the debug facility.
-msg <i>Message</i>	Defines the message that the snmptrap command will send. This value specifies the information the trap will hold. This information is in the text format. You must order this flag as the last one in the list when you specify this command.
-target <i>TargetHost</i>	Specifies the target network-manger host to which the trap message is sent. It is different from -host flag. If you do not specify the -host and -target flags, the trap is sent to the VIOS SNMP agent on the local host.

Exit Status

See "Exit status for Virtual I/O Server commands" on page 1.

Examples

To display the current system information and snmp configuration information, type the following command:

```
snmp_trap -mode hello world
```

Related Information

The **snmp_trap** command and the **cl_snmp** command.

snmpv3_ssw command

Purpose

Switch the symbolic links among the non-encrypted snmpdv3 agent, encrypted snmpdv3 agent, and the snmpdv1 agent.

Syntax

```
snmpv3_ssw [-e | -n | -1 ]
```

Description

The **snmpv3_ssw** command switches the symbolic links among the non-encrypted snmpdv3 agent and snmpdv1 agent. The **snmpv3_ssw** command then starts the newly chosen SNMP agent. You can choose which version of SNMP agent to run.

Flags

-e	Switches to the encrypted version of snmpdv3 agent.
-n	Switches to the non-encrypted version of snmpdv3 agent.
-1	Switches to the snmpdv1 agent.

Exit Status

Return code	Description
0	Success
1	Invalid flag, argument, or command failure

Examples

To switch to the encrypted version of snmpdv3 agent, enter the following command:

```
snmp3_ssw -e
```

startnetsvc command

Purpose

Starts the ndpd-host, telnet, ftp, xntpd, ssh, snmp, ldap, or cimserver daemon.

Syntax

```
startnetsvc [NetworkService][TracingSelection]
```

Description

The **startnetsvc** command starts the ndpd-host, telnet, ftp, xntpd, ssh, snmp, ldap or cimserver daemon. By enabling the network service daemon, users can use that service to connect to the Virtual I/O Server.

Parameters

NetworkService

Use the following values:

ndpd-host

Enables the **ndpd-host** daemon.

telnet Enables the **telnet** daemon.

ftp Enables the **ftp** daemon.

xntpd Enables the **xntpd** daemon.

ssh Enables the **ssh** daemon.

snmp Enables the **snmp** daemon.

ldap Enables the **ldap** daemon.

cimserver

Enables the **cimserver** daemon.

ALL Enables all service daemons.

Specifying **ALL** starts all services, but does not enable the **tracelog** or **errorlog** options.

TraceSelection

Use the following values:

tracelog

Sends CLI tracing information to the system log.

Attention: Enabling the **tracelog** or **errorlog** can make the system log grow quickly.

errorlog

Sends system error log information to the system log.

Attention: Enabling the **tracelog** or **errorlog** can make the system log grow quickly.

Note: A padmin user can use the **vi** command to view files sent to the system log.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

1. To start the **telnet** daemon, type the following command:

```
startnetsvc telnet
```
2. To start the **ftp** daemon, type the following command:

```
startnetsvc ftp
```
3. To start the **tracelog** option, type the following command:

```
startnetsvc tracelog
```
4. To start all service daemons, type the following command:

```
startnetsvc ALL
```
5. To start the **ndpd-host** daemon, type the following command:

```
startnetsvc ndpd-host
```

This command produces output similar to the following:

```
0513-059 The ndpd-host Subsystem has been started. Subsystem PID is 356522.
```

Related Information

The `mkcpi` command, the `hostname` command, the `stopnetsvc` command, the `cfgnagg` command, the `netstat` command, the `entstat` command, the `cfgnamesrv` command, the `hostmap` command, the `traceroute` command, the `ping` command, the `optimizenet` command, and the `vi` command.

startsvc command

Purpose

Starts the agent that is specified by the agent name.

Syntax

```
startsvc AgentName
```

Description

The `startsvc` command starts the specified agent. The agent name is case sensitive. Use the `lssvc` command to obtain a list of valid agent names. The agent determines the operations that occur while the start command is running. If you attempt to restart an agent that is already started, you will receive a message indicating that the agent has already been started.

Note: For the IBM TotalStorage Productivity Center agent, `TPC`, the `lssvc` command only displays the valid agent that is used by the `cfgsvc` command during configuration.

Agent names

The following agents can be managed by the Virtual I/O Server .

<code>DIRECTOR_agent</code>	Starts the IBM Systems Director agent.
<code>ITM_base</code>	Starts the specified IBM Tivoli Monitoring agent.
<code>ITM_premium</code>	The <code>ITM_base</code> , <code>ITM_premium</code> , and <code>ITM_cec</code> agents are IBM Tivoli Monitoring agents. These agents provide system information including I/O virtualization mappings and system utilization. The <code>ITM_cec</code> agent is only available to the Integrated Virtualization Manager. The agent name is case sensitive. The <code>ITM_base</code> and <code>ITM_premium</code> agents cannot run concurrently.
<code>ITM_cec</code>	
<code>TSM_base</code>	Starts the IBM Tivoli Storage Manager agent.
<code>ITUAM_base</code>	Starts the IBM Tivoli Usage and Accounting Manager agent.
<code>TPC</code>	Starts the TotalStorage Productivity Center agent.
	The <code>TPC_fabric</code> and <code>TPC_data</code> agents are valid IBM TotalStorage Productivity Center agents for the <code>startsvc</code> command.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

1. To start the `ITM_base` agent, type the following command:

```
startsvc ITM_base
```
2. To start the `ITUAM` base agent, type the following command:

```
startsvc ITUAM_base
```

Related Information

The `lssvc` command, the `cfgsvc` command, and the `stopsvc` command.

For more information about the various agents, see the following information:

- IBM Tivoli software and the Virtual I/O Server
- Configuring the IBM Tivoli agents and clients on the Virtual I/O Server
- IBM Systems Director software
- Configuring the IBM Systems Director agent

startsysdump command

Purpose

Starts a kernel dump to the primary dump device.

Syntax

`startsysdump`

Description

The `startsysdump` command provides a command line interface to start a kernel dump to the primary dump device. Any previous kernel dumps will be erased before the dump is created. During a kernel dump, the following values can be displayed on the three-digit terminal display as follows. The user will be required to run the `snap` command to obtain the system dump.

0c0	Indicates that the dump completed successfully.
0c1	Indicates that an I/O occurred during the dump.
0c2	Indicates that the dump is in progress.
0c4	Indicates that the dump is too small.
0c5	Indicates a dump internal error.
0c6	Prompts you to make the secondary dump device ready.
0c7	Indicates that the dump process is waiting for a response from the remote host.
0c8	Indicates that the dump was disabled. In this case, no dump device was designated in the system configuration object for dump devices. The <code>startsysdump</code> command halts, and the system continues running.
0c9	Indicates that a dump is in progress.
0cc	Indicates that the system switched to the secondary dump device after attempting a dump to the primary device.

Note: When the dump completes, the system reboots.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

1. To start a kernel dump, type:
`startsysdump`

starttrace command

Purpose

Records selected system events.

Syntax

```
starttrace [ -event Event[, Event ] ...]
```

Description

The **starttrace** command starts the **trace** daemon which configures a trace session and starts the collection of system events. The data collected by the trace function is recorded in the trace log. A report from the trace log can be generated with the **cattracerpt** command.

Flags

-event *Event*[,*Event*] Specifies the user-defined events for which you want to collect trace data. The Event list items should be separated by commas.
Note: The following events are used to determine the pid, the cpuid and the exec path name in the **cattracerpt** report:

- 106 DISPATCH
- 10C DISPATCH IDLE PROCESS
- 134 EXEC SYSTEM CALL
- 139 FORK SYSTEM CALL
- 465 KTHREAD CREATE

If any of these events is missing, the information reported by the **cattracerpt** command will be incomplete. When using the **-event** flag, you should include all these events in the *Event* list.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

1. To trace hook 234 and the hooks that will allow you to see the process names, enter:

```
starttrace -event 234,106,10C,134,139,465
```

Related Information

The stoptrace command and the cattracerpt command.

stopnetsvc command

Purpose

Disables the ndpd-host, telnet, ftp, xntpd, ssh, snmp, ldap, or cimserver daemon.

Syntax

```
stopnetsvc [NetworkService][TracingSelection]
```


Description

The **stopnetsvc** command stops the **ndpd-host**, **telnet**, **ftp**, **xntpd**, **ssh**, **snmp**, **ldap**, or **cimserver** daemon. By disabling a service daemon, users can prevent anyone from being able to connect through the associated network service. The **tracelog** or **errorlog** options stop sending CLI tracing information or system error log information to the system log.

Parameters

NetworkService

The following values can be used:

ndpd-host

Disables the **ndpd-host** daemon.

telnet Disables the **telnet** daemon.

ftp Disables the **ftp** daemon.

xntpd Disables the **xntpd** daemon.

ssh Disables the **ssh** daemon.

snmp Disables the **snmp** daemon.

ldap Disables the **ldap** daemon.

cimserver

Disables the **cimserver** daemon.

ALL Disables all service daemons, but does not disable the **tracelog** or **errorlog** option.

TraceSelection

The following values can be used:

tracelog

Stops sending CLI tracing information to the system log.

errorlog

Stops sending system error log information to the system log.

Note: A padmin user can use the **vi** command to view files sent to the system log.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

1. To disable the **telnet** daemon, type the following command:

```
stopnetsvc telnet
```
2. To disable the **ftp** daemon, type the following command:

```
stopnetsvc ftp
```
3. To disable the **ndpd-host** daemon, type the following command:

```
stopnetsvc ndpd-host
```

This command produces output similar to the following:

```
0513-044 The /usr/sbin/ndpd-host Subsystem was requested to stop.
```

4. To stop the **tracelog** option, type the following command:

```
stopnetsvc tracelog
```
5. To disable all service daemons, type the following command:

```
stopnetsvc ALL
```

Related Information

The **mktcpip** command, the **hostname** command, the **startnetsvc** command, the **cfgnagg** command, the **netstat** command, the **entstat** command, the **cfgnamesrv** command, the **hostmap** command, the **traceroute** command, the **ping** command, the **optimizenet** command, and the **vi** command.

stopsvc command

Purpose

Stops the agent that is specified by the agent name.

Syntax

```
stopsvc AgentName
```

Description

The **stopsvc** command stops the specified agent. The agent name is case sensitive. Use the **lssvc** command to obtain a list of valid agent names. The agent determines the operations to occur during the stop command. If you attempt to stop an agent that is already stopped, you will receive a message indicating that the agent has already been stopped.

Note: For the IBM TotalStorage Productivity Center agent, **TPC**, the **lssvc** command only displays the valid agent used by the **cfgsvc** command during configuration.

Agent names

The following agents can be managed by the Virtual I/O Server .

DIRECTOR_agent	Stops the IBM Systems Director agent.
ITM_base	Stops the specified IBM Tivoli Monitoring agent.
ITM_premium	The ITM_base , ITM_premium , and ITM_cec agents are IBM Tivoli Monitoring agents. These agents provide system information including I/O virtualization mappings and system utilization. The ITM_cec agent is only available to the Integrated Virtualization Manager. The agent name is case sensitive. The ITM_base and ITM_premium agents cannot run concurrently.
ITM_cec	
TSM_base	Stops the IBM Tivoli Storage Manager agent.
ITUAM_base	Stops the IBM Tivoli Usage and Accounting Manager agent.
TPC	Stops the TotalStorage Productivity Center agent.
	The TPC_fabric and TPC_data agents are valid IBM TotalStorage Productivity Center agents for the stopsvc command.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

1. To stop the **TPC_data** agent, type the following command:

```
stopsvc TPC_data
```
2. To stop the **TPC_fabric** agent, type the following command:

```
stopsvc TPC_fabric
```
3. To stop the **DIRECTOR_agent** agent, type the following command:

```
stopsvc DIRECTOR_agent
```

4. To stop the ITM_base agent, type the following command:

```
stopsvc ITM_base
```

5. To stop the ITUAM_base agent, type the following command:

```
stopsvc ITUAMM_base
```

Related Information

The **lssvc** command, the **cfgsvc** command, and the **startsvc** command.

For more information about the various agents, see the following information:

- Configuring the IBM Systems Director agent
- Configuring the IBM Tivoli agents and clients on the Virtual I/O Server
- IBM Systems Director software
- IBM Tivoli software and the Virtual I/O Server

stoptrace command

Purpose

Stops the trace function.

Syntax

```
stoptrace
```

Description

The **stoptrace** command ends a trace session.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

1. To end a trace session, type:

```
stoptrace
```

Related Information

The **starttrace** command and the **cattracerpt** command.

stty command

Purpose

Sets, resets, and reports workstation operating parameters.

Syntax

```
stty [ -a ] [ -g ] [ Options ]
```

Description

The **stty** command sets certain I/O options for the device that is the current standard input. This command writes output to the device that is the current standard output.

This version of the operating system uses the standard interface to control the terminals, maintaining a compatibility with POSIX and BSD interfaces. The **stty** command supports both POSIX and BSD compliant options, but the usage of POSIX options is strongly recommended. A list of obsolete BSD options, with the corresponding POSIX options, is also provided.

When you redirect standard input from a tty device by typing:

```
stty -a </dev/ttyx
```

the **stty** command (POSIX) will hang while waiting for the **open()** of that tty until the RS-232 carrier detect signal has been asserted. Exceptions to this rule occur if the **cllocal** or **forcedcd** (128-port only) option is set.

Flags

- 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.

Options

The **stty** command supports following categories of options:

- “Control modes”
- “Input modes” on page 373
- “Output modes” on page 373
- “Local modes” on page 374
- “Hardware flow control modes” on page 375
- “Control assignments” on page 375
- “Combination modes” on page 376
- “Window size” on page 377
- “Obsolete options” on page 377

Control modes

Control mode	Description
cllocal	Assumes a line without modem control.
-cllocal	Assumes a line with modem control.
cread	Enables the receiver.
-cread	Disables the receiver.
cstopb	Selects 2 stop bits per character.
-cstopb	Selects 1 stop bit per character.
cs5, cs6, cs7, cs8	Selects character size.
hup, hupcl	Hangs up dial-up connection on the last close.
-hup, -hupcl	Does not hang up dial-up connection on the last close.
parenb	Enables parity generation and detection.
-parenb	Disables parity generation and detection.
parodd	Selects odd parity.
-parodd	Selects even parity.
0	Hangs up phone line immediately.

Control mode

speed

Description

Sets the workstation input and output speeds to the specified *speed* number of bits per second. All speeds are not supported by all hardware interfaces. Possible values for *speed* are: 50, 75, 110, 134, 200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 19.2, 38400, 38.4, *exta*, and *extb*.

ispeed *speed*

Note: *exta*, 19200, and 19.2 are synonyms; *extb*, 38400, and 38.4 are synonyms.

Sets the workstation input speed to the specified *speed* number of bits per second. All speeds are not supported by all hardware interfaces, and all hardware interfaces do not support this option. Possible values for *speed* are the same as for the *speed* option.

ospeed *speed*

Sets the workstation output speed to the specified *speed* number of bits per second. All speeds are not supported by all hardware interfaces, and all hardware interfaces do not support this option. Possible values for *speed* are the same as for the *speed* option.

Input modes

Input mode

brkint

Signals INTR on break.

-brkint

Does not signal INTR on break.

icrnl

Maps CR to NL on input.

-icrnl

Does not map CR to NL on input.

ignbrk

Ignores BREAK on input.

-ignbrk

Does not ignore BREAK on input.

igncr

Ignores CR on input.

-igncr

Does not ignore CR on input.

ignpar

Ignores parity errors.

-ignpar

Does not ignore parity errors.

inlcr

Maps NL to CR on input.

-inlcr

Does not map NL to CR on input.

inpck

Enables parity checking.

-inpck

Disables parity checking.

istrip

Strips input characters to 7 bits.

-istrip

Does not strip input characters to 7 bits.

iuclc

Maps uppercase alphabetic characters to lowercase.

-iuclc

Does not map uppercase alphabetic characters to lowercase.

ixany

Allows any character to restart output.

-ixany

Allows only the START (the Ctrl-Q key sequence) to restart output.

ixoff

Sends START/STOP characters when the input queue is nearly empty/full.

-ixoff

Does not send START/STOP characters.

ixon

Enables START/STOP output control. Once START/STOP output control has been enabled, you can pause output to the workstation by pressing the Ctrl-S key sequence and resume output by pressing the Ctrl-Q key sequence.

-ixon

Disables START/STOP output control.

imaxbel

Echoes the BEL character and discards the last input character if input overflows.

-imaxbel

Discards all input if input overflows.

parmrk

Marks parity errors.

-parmrk

Does not mark parity errors.

Output modes

Output mode

bs0, **bs1**

Description

Selects style of delay for backspaces (**bs0** signifies no delay).

Output mode	Description
cr0 , cr1 , cr2 , cr3	Selects style of delay for CR characters (cr0 signifies no delay).
ff0 , ff1	Selects style of delay for form feeds (ff0 signifies no delay).
nl0 , nl1	Selects style of delay for NL characters (nl0 signifies no delay).
ofill	Uses fill characters for delays.
-ofill	Uses timing for delays.
ocrnl	Maps CR characters to NL characters.
-ocrnl	Does not map CR characters to NL characters.
olcuc	Maps lowercase alphabetic characters to uppercase on output.
-olcuc	Does not map lowercase alphabetic characters to uppercase on output.
onlcr	Maps NL characters to CR-NL characters.
-onlcr	Does not map NL characters to CR-NL characters.
onlret	On the terminal, NL performs the CR function.
-onlret	On the terminal, NL does not perform the CR function.
onocr	Does not output CR characters at column zero.
-onocr	Outputs CR characters at column zero.
opost	Processes output.
-opost	Does not process output; that is, ignores all other output options.
ofdel	Uses DEL characters for fill characters.
-ofdel	Uses NUL characters for fill characters.
tab0 , tab1 , tab2	Selects style of delay for horizontal tabs (tab0 signifies no delay).
tab3	Expands tab character to variable number of spaces.
vt0 , vt1	Selects style of delay for vertical tabs (vt0 signifies no delay).

Local modes

Local mode	Description
echo	Echoes every character typed.
-echo	Does not echo characters.
echoctl	Echoes control characters as ^X (Ctrl-X), where X is the character given by adding 100 octal to the code of the control character.
-echoctl	Does not echo control characters as ^X (Ctrl-X).
echoe	Echoes the ERASE character as the "backspace space backspace" string. Note: This mode does not keep track of column position, so you can get unexpected results when erasing such things as tabs and escape sequences.
-echoe	Does not echo the ERASE character, just backspace.
echok	Echoes a NL character after a KILL character.
-echok	Does not echo a NL character after a KILL character.
echoke	Echoes the KILL character by erasing each character on the output line.
-echoke	Just echoes the KILL character.
echonl	Echoes the NL character.
-echonl	Does not echo the NL character.
echoprt	Echoes erased characters backwards with / (slash) and \ (backslash).
-echoprt	Does not echo erased characters backwards with / (slash) and \ (backslash).
icanon	Enables canonical input (canonical input allows input-line editing with the ERASE and KILL characters).
-icanon	Disables canonical input.
iexten	Specifies that implementation-defined functions shall be recognized from the input data. Recognition of the following control characters requires iexten to be set: eol2 , dsusp , reprint , discard , werase , lnext . The functions associated with these modes also require iexten to be set: imaxbel , echoke , echoprt , and echoctl .
-iexten	Specifies that implementation-defined functions shall not be recognized from the input data.
isig	Enables the checking of characters against the special control characters INTR, SUSP and QUIT.

Local mode	Description
-isig	Disables the checking of characters against the special control characters INTR, SUSP and QUIT.
noflsh	Does not clear buffers after INTR, SUSP, or QUIT control characters.
-noflsh	Clears buffers after INTR, SUSP, or QUIT control characters.
pending	Causes any input that is pending after a switch from raw to canonical mode to be re-input the next time a read operation becomes pending or the next time input arrives. Pending is an internal state bit.
-pending	No text is pending.
tostop	Signals SIGTOU for background output.
-tostop	Does not signal SIGTOU for background output.
xcase	Echoes uppercase characters on input, and displays uppercase characters on output with a preceding \ (backslash).
-xcase	Does not echo uppercase characters on input.

Hardware flow control modes

These options are extensions to the X/Open Portability Guide Issue 4 standard.

Extension	Description
cdxon	Enables CD hardware flow control mode on output.
-cdxon	Disables CD hardware flow control mode on output.
ctsxon	Enables CTS hardware flow control mode on output.
-ctsxon	Disables CTS hardware flow control mode on output.
dtrxoff	Enables DTR hardware flow control mode on input.
-dtrxoff	Disables DTR hardware flow control mode on input.
rtsxoff	Enables RTS hardware flow control mode on input.
-rtsxoff	Disables RTS hardware flow control mode on input.

Control assignments

To assign a control character to a character string, type:

```
stty Control String
```

where the *Control* parameter may be the INTR, QUIT, ERASE, KILL, EOF, EOL, EOL2, START, STOP, SUSP, DSUSP, REPRINT, DISCARD, WERASE, LNEXT, MIN, or TIME character. (Use the MIN and TIME characters with the **-icanon** option.)

Note: The values for MIN and TIME are interpreted as integer values, not as character values.

The *String* parameter may be any single character such as c. An example of this control assignment is:

```
stty STOP c
```

Another way of assigning control characters is to enter a character sequence composed of a \^ (backslash, caret) followed by a single character. If the single character after the ^ (caret) is one of the characters listed in the ^c (caret c) column of the following table, the corresponding control character value will be set. For example, to assign the DEL control character by using the ? (question mark) character, type the string \^? (backslash, caret, question mark), as in:

```
stty ERASE \^?
```

^c	Value
a, A	<SOH>
b, B	<STX>

^c	Value
c, C	<ETX>
d, D	<EOT>
e, E	<ENQ>
f, F	<ACK>
g, G	<BEL>
h, H	<BS>
i, I	<HT>
j, J	<LF>
k, K	<VT>
l, L	<FF>
m, M	<CR>
n, N	<SO>
o, O	<SI>
p, P	<DLE>
q, Q	<DC1>
r, R	<DC2>
s, S	<DC3>
t, T	<DC4>
u, U	<NAK>
v, V	<SYN>
w, W	<ETB>
x, X	<CAN>
y, Y	
z, Z	<SUB>
[<ESC>
\	<FS>
]	<GS>
^	<RS>
_	<US>
?	
@	<NUL>

Combination modes

Combination mode	Description
cooked	See the -raw option.
ek	Sets ERASE and KILL characters to the Ctrl-H and Ctrl-U key sequences, respectively.
evenp	Enables parenb and cs7 .
-evenp	Disables parenb and sets cs8 .
lcase, LCASE	Sets xcase , iuclc , and olcuc . Used for workstations with uppercase characters only.
-lcase, -LCASE	Sets -xcase , -iuclc , and -olcuc .
nl	Sets -icrnl and -onlcr .

Combination mode	Description
-nl	Sets icrnl , onlcr , -inlcr , -igncr , -ocrnl , and -onlret .
oddp	Enables parenb , cs7 , and parodd .
-oddp	Disables parenb and sets cs8 .
parity	See the evenp option.
-parity	See the -evenp option.
sane	Resets parameters to reasonable values.
raw	Allows raw mode input (no input processing, such as erase, kill, or interrupt); parity bit passed back.
-raw	Allows canonical input mode.
tabs	Preserves tabs.
-tabs, tab3	Replaces tabs with spaces when printing.

Window size

Window size	Description
cols <i>n</i>, columns <i>n</i>	The terminal (window) size is recorded as having <i>n</i> columns.
rows <i>n</i>	The terminal (window) size is recorded as having <i>n</i> rows.
size	Prints the terminal (window) sizes to standard output (first rows and then columns).

Obsolete options

The following BSD options are supported by the **stty** command. For each of them, the recommended POSIX option is given.

Option	Description
all	Use the stty -a command to display all current settings.
crt	Use the sane option to reset parameters to reasonable values.
crtbs	Use the -echoe option.
crterase	Use the echoe option.
-crterase	Use the -echoe option.
crtkill	Use the echoke option.
-crtkill	Use the echok and -echoke options.
ctlecho	Use the echoctl option.
-ctlecho	Use the -echoctl option.
decctlq	Use the -ixany option.
-decctlq	Use the ixany option.
even	Use the evenp option.
-even	Use the -evenp option.
everything	Use the stty -a command to display all current settings.
litout	Use the -opost option.
-litout	Use the opost option.
odd	Use the oddp option.
-odd	Use the -oddp option.
pass8	Use the -istrip option.
-pass8	Use the istrip option.
prterase	Use the echopr option.
speed	Use the stty command to display current settings.
tandem	Use the ixoff option.
-tandem	Use the -ixoff option.

Examples

1. To display a short listing of your workstation configuration, type:

```
stty
```

This lists settings that differ from the defaults.

2. To display a full listing of your workstation configuration, type:

```
stty -a
```

3. To enable a key sequence that stops listings from scrolling off the screen, type:

```
stty ixon ixany
```

This sets **ixon** mode, which lets you stop runaway listing by pressing the Ctrl-S key sequence. The **ixany** flag allows you to resume the listing by pressing any key. The normal workstation configuration includes the **ixon** and **ixany** flags, which allows you to stop a listing with the Ctrl-S key sequence that only the Ctrl-Q key sequence will restart.

4. To reset the configuration after it has been messed up, type:

```
Ctrl-J stty sane Ctrl-J
```

Press the Ctrl-J key sequence before and after the command instead of the Enter key. The system usually recognizes the Ctrl-J key sequence when the parameters that control Enter key processing are messed up.

Sometimes the information displayed on the screen may look strange, or the system will not respond when you press the Enter key. This can happen when you use the **stty** command with parameters that are incompatible or that do things you don't understand. It can also happen when a screen-oriented application ends abnormally and does not have a chance to reset the workstation configuration.

Entering the **stty sane** command sets a reasonable configuration, but it may differ slightly from your normal configuration.

5. To save and restore the terminal's configuration:

```
OLDCONFIG=`stty -g`      # save configuration
stty -echo              # do not display password
echo "Enter password: \c"
read PASSWD             # get the password
stty $OLDCONFIG         # restore configuration
```

This command saves the workstation's configuration, turns off echoing, reads a password, and restores the original configuration.

Entering the **stty -echo** command turns off echoing, which means that the password does not appear on the screen when you type it at the keyboard. This action has nothing to do with the **echo** command, which displays a message on the screen.

File

`/usr/bin/stty` Contains the **stty** command.

su command

Purpose

Changes the user ID that is associated with a session.

Syntax

su [-] [Name [Argument...]]

Description

The **su** command changes user credentials to those of the root user or to the user specified by the Name parameter, and then initiates a new session.

Any arguments, such as flags or parameters, that are specified by the Arguments parameter must relate to the login shell defined for the user specified by the Name parameter. These arguments are passed to the specified user's login shell. For example, if the login shell for user Fred is /usr/bin/rksh.

The following functions are performed by the **su** command:

account checking

Validates the user account to be certain it exists, that it is enabled for the **su** command, that the current user is in a group permitted to switch to this account with the **su** command, and that it can be used from the current controlling terminal.

user authentication

Validates the user's identity, using the system-defined primary authentication methods for the user. If a password has expired, the user must supply a new password.

credentials establishment

Establishes initial user credentials, using the values in the user database. These credentials define the user's access rights and accountability on the system.

session initiation

If the - flag is specified, the **su** command initializes the user environment from the values in the user database and the /etc/environment file. When the - flag is not used, the **su** command does not change the directory.

These functions are performed in the sequence shown. If one function is unsuccessful, the succeeding functions are not done.

To restore the previous session, type exit or press the Ctrl-D key sequence. This action ends the shell called by the **su** command and returns you to the previous shell, user ID, and environment.

Each time the **su** command is executed, an entry is made in the /var/adm/sulog and /home/ios/logs/sulog file. Both log files record the following information: date, time, system name, and login name. Both log files also records whether or not the login attempt was successful: a plus sign indicates a successful login, and a minus sign indicates an unsuccessful login.

Note: Successful use of the **su** command resets the unsuccessful login count attribute in the /etc/security/lastlog file only if the user's rlogin and login attributes are both set to false in /etc/security/user. Otherwise, the **su** command does not reset the unsuccessful login count, because the administrator often uses the **su** command to fix user account problems. The user is able to reset the attribute through a local or remote login.

Flags

- Specifies that the process environment is to be set as if the user had logged in to the system using the login command.

Name Tip: Nothing in the current environment is propagated to the new shell.
Specifies user ID.

Arguments Specifies the clear text password for the bindDN that is used to bind to the LDAP server.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Files

/home/ios/logs/sulog Contains the **su** command log files.
Note: A padmin user can use the **vi** command to view log files.

Examples

To change the user ID associated to a session, type the following command:

```
su - bob
```

Related Information

The **vi** command.

svmon command

Purpose

Captures and analyzes a snapshot of virtual memory.

Syntax

Global Report

```
svmon [-G [-i Intvl [ NumIntvl ]][-z]]
```

Process Report

```
svmon[-P [pid1...pidn] [-r] [-u | -g | -v] [-ns] [-wfc] [-q[s|m|L|S]] [-t Count] [ -i Intvl [NumIntvl] ] [-l] [-j] [-z] [-m] ]
```

Segment Report

```
svmon [-S [sid1...sidn] [-r] [-u | -P | -g | -v] [-ns] [-wfc] [-g [s|m|L|S]] [-t Count] [ -i Intvl [NumIntvl] ] [-l] [-j] [-z] [-m] ]
```

Detailed Report

```
svmon [-D sid1...sidn [-b] [-q [s|m|L|S]] [ -i Intvl [NumIntvl] ]][-z]]
```

Framed Report

```
svmon [-F [fr1...frn] [-q [s|m|L|S]] [-i Intvl [NumIntvl] ]][-z]] ]
```

Command Report

svmon [-C *cmd1...cmdn*] [-r] [-u | -p | -g | -v] [-ns] [-wfc] [-q [*s | m | L | S*]] [-t *Count*] [-i *Intvl* [*NumIntvl*]] [-d] [-l] [-j] [-z] [-m]]

User Report

svmon [-U [*lognm1...lognmn*]] [-r] [-u | -p | -g | -v] [-ns] [-wfc] [-t *Count*] [-i *Intvl* [*NumIntvl*]] [-d] [-l] [-j] [-z] [-m]]

Workload Management Class Report

svmon [-W [*class1...classn*]] [-e] [-r]] [-u | -p | -g | -v] [-ns] [-wfc] [-q [*s | m | L | S*]] [-t *Count*] [-i *Intvl* [*NumIntvl*]] [-d] [-l] [-j] [-z] [-m]]

Workload Management Tier Report

svmon [-T [*tier1...tiern*]] [-a *superclass*] [-x] -e [-r] [-u | -p | -g | -v] [-ns] [-wfc] [-q [*s | m | L | S*]] [-t *Count*] [-i *Intvl* [*NumIntvl*]] [-d] [-l] [-j] [-z] [-m]]

Description

Flags

If no command line flag is given, then the **-G** flag is the default.

-a <i>SupClassName</i>	Restricts the scope to the subclasses of the <i>SupClassName</i> class parameter (in the Tier report -T). The parameter is a superclass name. No list of class is supported.
-b	Shows the status of the reference and modified bits of all the displayed frames (detailed report -D). When shown, the reference bit of the frame is reset. When used with the -i flag, it detects which frames are accessed between each interval. Note: Use this flag with caution because of its impact on performance.
-C <i>Command1...CommandN</i>	Displays memory usage statistics for the processes running the command name <i>Commandnm1...CommandnmN</i> . <i>Commandnm</i> is a string. It is the exact basename of an executable file.
-d	Displays for a given entity, the memory statistics of the processes belonging to the entity (user name or command name).
-D <i>SID1...SIDN</i>	Displays memory-usage statistics for segments <i>SID1...SIDN</i> , and a detail status of all frames of each segment. Segment ids specified must be of primary segments.
-e	Displays the memory usage statistics of the subclasses of the <i>Class</i> parameter in the Workload Class report -W and in the Tier report -T. The class parameter of -W or -a needs to be a superclass name.
-f	Indicates that only persistent segments (files) are to be included in the statistics. By default all segments are analyzed.
-F [<i>Frame1...FrameN</i>]	Displays the status of frames <i>Frame1...FrameN</i> including the segments to which they belong. If no list of frames is supplied, the percentage of memory used displays.
-g	Indicates that the information to be displayed is sorted in decreasing order by the total number of pages reserved or used on paging space. This flag, in conjunction with the segment report, shifts the non-working segment at the end of the sorted list
-G	Displays a global report.
-i <i>Interval</i> [<i>NumIntervals</i>]	Instructs the svmon command to display statistics repetitively. Statistics are collected and printed every <i>Interval</i> seconds. <i>NumIntervals</i> is the number of repetitions; if not specified, svmon runs until user interruption, Ctrl-C. Note: It may take a few seconds to collect statistics for some options. The observed interval may be larger than the specified interval.

- j** Shows, for each persistent segment, the file path referred.
Note: This flag should be used with caution because of its potential performance impacts (especially with **svmon -S**).
 - l** Shows, for each displayed segment, the list of process identifiers that use the segment and, according to the type of report, the entity name (login, command, tier, or class) to which the process belongs. For special segments, a label is displayed instead of the list of process identifiers.
- System segment**
This label is displayed for segments that are flagged *system*.
- Unused segment**
This label is displayed for segments which are not used by any existing processes. For example, persistent segments relative to files no longer in usage.
- Shared library text**
This label is displayed for segments that contain text of shared library, and that may be used by most of the processes (for example, *libc.a*). This is to prevent the display of a long list of processes.
- m** Displays information about source segment and mapping segment when a segment is mapping a source segment. The default is to display only information about the mapping segment.
 - n** Indicates that only non-system segments are to be included in the statistics. By default, all segments are analyzed.
 - s** Indicates that only system segments are to be included in the statistics. By default, all segments are analyzed.
 - p** Indicates that the information to be displayed is sorted in decreasing order by the total number of pages pinned.
 - P [PID1...PIDN]** Displays memory usage statistics for process *PID1...PIDN*. PID is a decimal value. If no list of process IDs (PIDs) is supplied, memory usage statistics are displayed for all active processes.
 - q [small | Large]** Filters results regarding whether they deal with pages of the requested size or not. The requested page size is specified through the option sub-argument. 4KB pages are specified with *s* and 16MB pages with *Large*. If no sub-argument is specified, the default is 16MB pages (*Large*). Metrics per page size are also displayed.
 - r** Displays the ranges within the segment pages which have been allocated. A working segment may have two ranges because pages are allocated by starting from both ends and moving towards the middle.
 - S [SID1...SIDN]** Displays memory-usage statistics for segments *SID1...SIDN*. SID is a hexadecimal value. Segment IDs specified must be of primary segments. If no list of segment IDs (SIDs) is supplied, memory usage statistics are displayed for all defined segments.
 - t Count** Displays memory usage statistics for the top *Count* object to be printed
 - T [Tier1...TierN]** Displays memory usage statistics of all the classes of the tier numbers *Tier1...TierN*. If no list of tier is supplied, memory usage statistics displayed for all the defined tiers.
 - u** Indicates that the information to be displayed is sorted in decreasing order by the total number of pages in real memory. It is the default sorting criteria if none of the following flags are present: **-p**, **-g** and **-v**.
 - U [LogName1...LogNameN]** Displays memory usage statistics for the login name *LogName1...LogNameN*. *LogName* is a string, it is an exact login name. If no list of login identifier is supplied, memory usage statistics are displayed for all defined login identifiers.
 - v** Indicates that the information to be displayed is sorted in decreasing order by the total number of pages in virtual space. This flag, in conjunction with the segment report, shifts the non-working segment at the end of the sorted list.
 - w** Indicates that only working segments are to be included in the statistics. By default all segments are analyzed.

-W [<i>Clmn1...ClmnN</i>]	Displays memory usage statistics for the workload management class <i>Clmn1...ClmnN</i> . <i>Clmn</i> is a string. It is the exact name of a class. For a subclass, the name should have the form <i>superclassname.subclassname</i> . If no list of class name is supplied, memory usage statistics display for all defined class names.
-c	Indicates that only client segments are to be included in the statistics. By default, all segments are analyzed.
-x	Displays memory usage statistics for the segments for every class of a tier in the Tier report -T.
-z	Displays the maximum memory size dynamically allocated by svmon during its execution.

Examples

To print out global statistics, type :

```
svmon -G
      size      inuse      free      pin      virtual
memory 262144    99728    162416    52690    81741
pg space 393216      706
      work      pers      clnt
pin    52690      0        0
in use 81741      0        17987
```

Related Information

Logical volume storage and device management

syncvg command

Purpose

Synchronizes logical volume copies that are not current.

Syntax

```
syncvg { -lv | -pv | -vg } Name ...
```

Description

The **syncvg** command synchronizes the physical partitions, which are copies of the original physical partition, that are not current. The **syncvg** command can be used with logical volumes, physical volumes, or volume groups, with the *Name* parameter representing the logical volume name, physical volume name, or volume group name.

Unless disabled, the copies within a volume group are synchronized automatically when the volume group is activated by the **activatevg** command.

Note: For the **syncvg** command to be successful, at least one good copy of the logical volume should be accessible, and the physical volumes that contains this copy should be in ACTIVE state.

The **syncvg** command will check for the *NUM_PARALLEL_LPS* environment variable. The value of *NUM_PARALLEL_LPS* will be used to set the number of logical partitions to be synchronized in parallel.

Flags

-lv	Specifies that the <i>Name</i> parameter represents a logical volume device name.
------------	---

-pv	Specifies that the <i>Name</i> parameter represents a physical volume device name.
-vg	Specifies that the <i>Name</i> parameter represents a volume group device name.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

- To synchronize the copies on physical volumes **hdisk04** and **hdisk05**, type:

```
syncvg -pv hdisk04 hdisk05
```
- To synchronize the copies on volume groups **vg04** and **vg05**, type:

```
syncvg -vg vg04 vg05
```

Related Information

The **mkvg** command, the **chvg** command, the **extendvg** command, the **lsvg** command, the **mirrorios** command, the **unmirrorios** command, the **activatevg** command, the **deactivatevg** command, the **importvg** command, the **exportvg** command, and the **reducevg** command.

sysstat command

Purpose

Displays a summary of current system activity.

Syntax

```
sysstat [ -long | -short ] [ User]
```

Description

The **sysstat** command provides the following information: time of day, amount of time since last system startup, number of users logged in and number of processes running.

Flags

-long	Prints the summary in long form. This is the default
-short	Prints the time of day, amount of time since last system startup, number of Users logged in and number of processes running.

Parameters

<i>User</i>	Specify an existing user name.
-------------	--------------------------------

tail command

Purpose

Displays the last few lines of a file.

Syntax

Standard Syntax

```
tail [ -f ] [ -c Number | -n Number | -m Number | -b Number | -k Number ] [ File ]
```

To Display Lines in Reverse Order

```
tail [ -r ] [ -n Number ] [ File ]
```

Description

The **tail** command writes the file specified by the *File* parameter to standard output beginning at a specified point. If no file is specified, standard input is used. The *Number* variable specifies how many units to write to standard output. The value for the *Number* variable can be a positive or negative integer. If the value is preceded by + (plus sign), the file is written to standard output starting at the specified number of units from the beginning of the file. If the value is preceded by - (minus sign), the file is written to standard output starting at the specified number of units from the end of the file. If the value is not preceded by + (plus sign) or - (minus sign), the file is read starting at the specified number of units from the end of the file.

The type of unit used by the *Number* variable to determine the starting point for the count is determined by the **-b**, **-c**, **-k**, **-m**, or **-n** flag. If one of these flags is not specified, the **tail** command reads the last ten lines of the specified file and writes them to standard output. This is the same as entering **-n 10** at the command line.

The **-m** flag provides consistent results in both single- and double-byte character environments. The **-c** flag should be used with caution when the input is a text file containing multibyte characters, because output can be produced that does not start on a character boundary.

Flags

-b <i>Number</i>	Reads the specified file beginning at the 512-byte block location indicated by the <i>Number</i> variable.
-c <i>Number</i>	Reads the specified file beginning at the byte location indicated by the <i>Number</i> variable.
-f	If the input file is a regular file or if the <i>File</i> parameter specifies a FIFO (first-in-first-out), the tail command does not terminate after the last specified unit of the input file has been copied, but continues to read and copy additional units from the input file as they become available. If no <i>File</i> parameter is specified and standard input is a pipe, the -f flag is ignored. The tail -f command can be used to monitor the growth of a file being written by another process.
-k <i>Number</i>	Reads the specified file beginning at the 1KB block location indicated by the <i>Number</i> variable.
-m <i>Number</i>	Reads the specified file beginning at the multibyte character location indicated by the <i>Number</i> variable. Using this flag provides consistent results in both single- and double-byte character-code-set environments.
-n <i>Number</i>	Reads the specified file from the first or last line location as indicated by the sign (+ or - or none) of the <i>Number</i> variable and offset by the number of lines <i>Number</i> .
-r	Displays the output from the end of the file in reverse order. The default for the -r flag prints the entire file in reverse order. If the file is larger than 20,480 bytes, the -r flag displays only the last 20,480 bytes.

The **-r** flag is valid only with the **-n** flag. Otherwise, it is ignored.

Exit Status

This command returns the following exit values:

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

Examples

1. To display the last 10 lines of the notes file, enter:

```
tail notes
```

2. To specify the number of lines to start reading from the end of the notes file, enter:

```
tail -n 20 notes
```

3. To display the notes file a page at a time, beginning with the 200th byte, enter:

```
tail -c +200 notes | pg
```

4. To follow the growth of a file, enter:

```
tail -f accounts
```

This displays the last 10 lines of the accounts file. The **tail** command continues to display lines as they are added to the accounts file. The display continues until you press the Ctrl-C key sequence to stop it.

File

`/usr/bin/tail` Contains the **tail** command.

Related Information

The **head** command and the **more** command

tee command

Purpose

Displays the output of a program and copies it into a file.

Syntax

```
tee [ -a ] [ -i ] [ File ... ]
```

Description

The **tee** command reads standard input, then writes the output of a program to standard output and simultaneously copies it into the specified file or files.

Flags

-a Adds the output to the end of *File* instead of writing over it.
-i Ignores interrupts.

Exit Status

This command returns the following exit values:

0 The standard input was successfully copied to all output files.
>0 An error occurred.

Note: If a write to any successfully opened *File* operand is not successful, writes to other successfully opened *File* operands and standard output will continue, but the exit value will be >0.

Examples

1. To view and save the output from a command at the same time:

```
lint program.c | tee program.lint
```

This displays the standard output of the command **lint program.c** at the workstation, and at the same time saves a copy of it in the file `program.lint`. If a file named `program.lint` already exists, it is deleted and replaced.

2. To view and save the output from a command to an existing file:

```
lint program.c | tee -a program.lint
```

This displays the standard output of the **lint program.c** command at the workstation and at the same time appends a copy of it to the end of the `program.lint` file. If the `program.lint` file does not exist, it is created.

Files

`/usr/bin/tee` Contains the **tee** command.

topas command

Purpose

Reports selected local system statistics.

Syntax

```
topas [ -cpus number_of_monitored_hot_CPUs ] [ -disks number_of_monitored_hot_disks ] [ -interval monitoring_interval_in_seconds ] [ -nets number_of_monitored_hot_network_interfaces ] [ -procs number_of_monitored_hot_processes ] [ -wlms number_of_monitored_hot_WLM_classes ] [ -procsdisp | -wlmdisp | -cecdisp ]
```

Description

The **topas** command reports selected statistics about the activity on the local system. The command displays its output in a format suitable for viewing on an 80x25 character-based display.

If the **topas** command is invoked without flags, it runs as if invoked with the following command line:

```
topas -disks 20 -interval 2 -nets 20 -procs 20 -wlms 20 -cpus 20
```

The program extracts statistics from the system with an interval specified by the *monitoring_interval_in_seconds* argument. The default output, as shown below, consists of two fixed parts and a variable section. The top two lines at the left of the display show the name of the system the **topas** command runs on, the date and time of the last observation, and the monitoring interval.

The second fixed part fills the rightmost 25 positions of the display. It contains five subsections of statistics, as follows:

EVENTS/QUEUES

Displays the per-second frequency of selected system-global events and the average size of the thread run and wait queues:

Cswitch

The number of context switches per second over the monitoring interval.

Syscalls

The total number of system calls per second executed over the monitoring interval.

Reads The number of read system calls per second executed over the monitoring interval.

Writes The number of write system calls per second executed over the monitoring interval.

Forks The number of fork system calls per second executed over the monitoring interval.

Execs The number of exec system calls per second executed over the monitoring interval.

Runqueue

The average number of threads that were ready to run but were waiting for a processor to become available.

Waitqueue

The average number of threads that were waiting for paging to complete.

FILE/TTY

Displays the per-second frequency of selected file and tty statistics.

Readch The number of bytes read per second through the **read** system call over the monitoring interval.

Writech

The number of bytes written per second through the **write** system call over the monitoring interval.

Rawin The number of raw bytes read per second from TTYs over the monitoring interval.

Ttyout The number of bytes written to TTYs per second over the monitoring interval.

Igets The number of calls per second to the inode lookup routines over the monitoring interval.

Namei The number of calls per second to the pathname lookup routines over the monitoring interval.

Dirblk The number of directory blocks scanned per second by the directory search routine over the monitoring interval.

PAGING

Displays the per-second frequency of paging statistics.

Faults Total number of page faults taken per second over the monitoring interval. This includes page faults that do not cause paging activity.

Steals Physical memory 4K frames stolen per second by the virtual memory manager over the monitoring interval.

PgspIn Number of 4K pages read from paging space per second over the monitoring interval.

PgspOut

Number of 4K pages written to paging space per second over the monitoring interval.

PageIn Number of 4K pages read per second over the monitoring interval. This includes paging activity associated with reading from file systems. Subtract **PgspIn** from this value to get the number of 4K pages read from file systems per second over the monitoring interval.

PageOut

Number of 4K pages written per second over the monitoring interval. This includes paging activity associated with writing to file systems. Subtract **PgspOut** from this value to get the number of 4K pages written to file systems per second over the monitoring interval.

Sios The number of I/O requests per second issued by the virtual memory manager over the monitoring interval.

MEMORY

Displays the real memory size and the distribution of memory in use.

Real,MB

The size of real memory in megabytes.

% Comp

The percentage of real memory currently allocated to computational page frames. Computational page frames are generally those that are backed by paging space.

% Noncomp

The percentage of real memory currently allocated to non-computational frames. Non-computational page frames are generally those that are backed by file space, either data files, executable files, or shared library files.

% Client

The percentage of real memory currently allocated to cache remotely mounted files.

PAGING SPACE

Display size and utilization of paging space.

Size,MB

The sum of all paging spaces on the system, in megabytes.

% Used

The percentage of total paging space currently in use.

% Free The percentage of total paging space currently free.

NFS

Display NFS stats in calls per second

- Server V2 calls/sec
- Client V2 calls/sec
- Server V3 calls/sec
- Client V3 calls/sec

The variable part of the **topas** display can have one, two, three, four, or five subsections. If more than one subsection displays, they are always shown in the following order:

- CPU

- Network Interfaces
- Physical Disks
- WorkLoad Management Classes
- Processes

When the **topas** command is started, it displays all subsections for which hot entities are monitored. The exception to this is the WorkLoad Management (WLM) Classes subsection, which is displayed only when WLM is active.

CPU Utilization

This subsection displays a bar chart showing cumulative CPU usage. If more than one CPU exists, a list of CPUs can be displayed by pressing the **c** key *twice*. Pressing the **c** key only once will turn this subsection off. The following fields are displayed by both formats:

User% This shows the percent of CPU used by programs executing in user mode. (Default sorted by User%)

Kern% This shows the percent of CPU used by programs executing in kernel mode.

Wait% This shows the percent of time spent waiting for IO.

Idle% This shows the percent of time the CPU(s) is idle.

When this subsection first displays the list of hot CPUs, the list is sorted by the User% field. However, the list can be sorted by the other fields by moving the cursor to the top of the desired column.

Network Interfaces

This subsection displays a list of hot network interfaces. The maximum number of interfaces displayed is the number of hot interfaces being monitored, as specified with the **-nets** flag. A smaller number of interfaces will be displayed if other subsections are also being displayed. Pressing the **n** key turns off this subsection. Pressing the **n** key again shows a one-line report summary of the activity for all network interfaces. Both reports display the following fields:

Interf The name of the network interface.

KBPS The total throughput in megabytes per second over the monitoring interval. This field is the sum of kilobytes received and kilobytes sent per second.

I-Pack The number of data packets received per second over the monitoring interval.

O-Pack The number of data packets sent per second over the monitoring interval.

KB-In The number of kilobytes received per second over the monitoring interval.

KB-Out
The number of kilobytes sent per second over the monitoring interval.

When this subsection first displays the list of hot network interfaces, the list is sorted by the KBPS field. However, the list can be sorted by the other fields by moving the cursor to the top of the desired column. Sorting is only valid for up to 16 network adapters.

Physical Disks

This subsection displays a list of hot physical disks. The maximum number of physical disks displayed is the number of hot physical disks being monitored as specified with the **-disks** flag. A smaller number of physical disks will be displayed if other subsections are also being displayed. Pressing the **d** key turns off this subsection. Pressing the **d** key again shows a one-line report summary of the activity for all physical disks. Both reports display the following fields:

Disk The name of the physical disk.

Busy% Indicates the percentage of time the physical disk was active (bandwidth utilization for the drive).

KBPS The number of kilobytes read and written per second over the monitoring interval. This field is the sum of **KB-Read** and **KB-Writ**.

TPS The number of transfers per second that were issued to the physical disk. A transfer is an I/O request to the physical disk. Multiple logical requests can be combined into a single I/O request to the disk. A transfer is of indeterminate size.

KB-Read

The number of kilobytes read per second from the physical disk.

K -Writ

The number of kilobytes written per second to the physical disk.

When this subsection first displays the list of hot physical disks, the list is sorted by the KBPS field. However, the list can be sorted by the other fields by moving the cursor to the top of the desired column. Sorting is only valid for up to 128 physical disks.

WLM Classes

This subsection displays a list of hot WorkLoad Management (WLM) Classes. The maximum number of WLM classes displayed is the number of hot WLM classes being monitored as specified with the **-wlmdisp** flag. A smaller number of classes will be displayed if other subsections are also being displayed. Pressing the **w** key turns off this subsection. The following fields are displayed for each class:

% CPU Utilization

The average CPU utilization of the WLM class over the monitoring interval.

% Mem Utilization

The average memory utilization of the WLM class over the monitoring interval.

% Blk I/O

The average percent of Block I/O of the WLM class over the monitoring interval.

When this subsection first displays the list of hot WLM classes, the list will be sorted by the CPU% field. However, the list can be sorted by the other fields by moving the cursor to the top of the desired column.

Processes

This subsection displays a list of hot processes. The maximum number of processes displayed is the number of hot processes being monitored as specified with the **-procs** flag. A smaller number of processes will be displayed if other subsections are also being displayed. Pressing the **p** key turns off this subsection. The process are sorted by their CPU usage over the monitoring interval. The following fields are displayed for each process:

Name The name of the executable program executing in the process. The name is stripped of any pathname and argument information and truncated to 9 characters in length.

Process ID
The process ID of the process.

% CPU Utilization
The average CPU utilization of the process over the monitoring interval. The first time a process is shown, this value is the average CPU utilization over the lifetime of the process.

Paging Space Used
The size of the paging space allocated to this process. This can be considered an expression of the footprint of the process but does not include the memory used to keep the executable program and any shared libraries it may depend on.

Process Owner (if the WLM section is off)
The user name of the user who owns the process.

WorkLoad Management (WLM) Class (if the WLM section is on)
The WLM class to which the process belongs.

Implementation Specifics

Changes to WLM that are shown by **topas** (like adding new classes, or changing existing class names) will not be reflected after starting **topas**. You must stop **topas** and all clients which use **Spmi**, then restart after the WLM changes are made. This is also the case for Disks and Network Adapters added after **topas** or any other **Spmi** consumer is started.

Sample Default Output

The following is an example of the display generated by the **topas** command:

```
Topas Monitor for host:  niller          EVENTS/QUEUES  FILE/TTY
Mon Mar 13 15:56:32 2000  Interval:  2    Cswitch      113  Readch  1853576
                               Syscall     2510  Writch   49883
CPU  User%  Kern%  Wait%  Idle%  Reads      466  Rawin    0
cpu0  7.0    4.0    0.0    89.0   Writes     12  Ttyout   706
cpu1  1.0    8.0    0.0    91.0   Forks      0  Igets    0
cpu2  0.0    0.0    0.0    100.0  Execs      0  Namei    0
                               Runqueue    0.0  Dirblk   0
                               Waitqueue   0.0
Interf  KBPS  I-Pack  O-Pack  KB-In  KB-Out  PAGING
lo0     100.4  45.7    45.7    50.2    50.2    Faults
tr0      2.0    4.4     3.4     1.4     0.6    Steals
Disk  Busy%  KBPS    TPS  KB-Read  KB-Writ  PgspIn  0  Real,MB  255
hdisk0  0.0    0.0    0.0    0.0    0.0    PgspOut  0  % Comp   81.0
hdisk1  0.0    0.0    0.0    0.0    0.0    PageIn   0  % Noncomp 19.0
                               PageOut   0  % Client  3.0
WLM-Class (Active) CPU%  Mem%  Disk%  Sios  0  PAGING SPACE
System          8    41    12    0    0  Size,MB  0
Shared         1    24    9    0    0  % Used
                               % Free
```


Name	PID	CPU%	PgSP	Class	NFS	calls/sec
topas	(35242)	3.0	0.3	System	ServerV2	0 Press:
X	(3622)	1.4	44.4	System	ClientV2	0 "h" for help.
notes	(25306)	1.3	123.3	System	ServerV3	0 "q" to quit.
					ClientV3	0

Sample Full Screen Process Output

```
Topas Monitor for host: mothra Interval: 2 Wed Nov 8 12:27:34 2000
PGFAULTS
USER PID PPID PRI NI DATA TEXT PAGE TIME CPU% I/O OTH COMMAND
root 1806 0 37 41 16 3374 16 13:25 1.0 0 0 gil
root 1032 0 16 41 3 3374 3 0:00 0.0 0 0 lrud
root 1290 0 60 41 4 3374 4 0:02 0.0 0 0 xmgc
root 1548 0 36 41 4 3374 4 0:26 0.0 0 0 netm
root 1 0 60 20 197 9 180 0:24 0.0 0 0 init
root 2064 0 16 41 4 3374 4 0:04 0.0 0 0 wlmsched
root 2698 1 60 20 14 2 14 0:00 0.0 0 0 shlap
root 3144 1 60 20 40 1 36 5:19 0.0 0 0 syncd
root 3362 0 60 20 4 3374 4 0:00 0.0 0 0 lvmbb
root 3666 1 60 20 135 23 123 0:00 0.0 0 0 errdemon
root 3982 0 60 20 4 3374 4 0:01 0.0 0 0 rtcmd
root 4644 1 17 20 6 3374 6 0:00 0.0 0 0 dog
root 4912 1 60 20 106 13 85 0:00 0.0 0 0 srcmstr
root 5202 4912 60 20 94 8 84 0:01 0.0 0 0 syslogd
root 5426 4912 60 20 195 76 181 0:12 0.0 0 0 sendmail
root 5678 4912 60 20 161 11 147 0:01 0.0 0 0 portmap
root 5934 4912 60 20 103 11 88 0:00 0.0 0 0 inetd
root 6192 4912 60 20 217 61 188 0:21 0.0 0 0 snmpd
root 6450 4912 60 20 137 10 116 0:00 0.0 0 0 dpid2
root 6708 4912 60 20 157 29 139 0:06 0.0 0 0 hostmibd
root 0 0 16 41 3 3374 3 7:08 0.0 0 0
root 6990 1 60 20 106 10 86 0:06 0.0 0 0 cron
```

Sample Full-Screen WorkLoad Management Classes Output

```
Topas Monitor for host: mothra Interval: 2 Wed Nov 8 12:30:54 2000
WLM-Class (Active) CPU% Mem% Disk-I/O%
System 0 0 0
Shared 0 0 0
Default 0 0 0
Unmanaged 0 0 0
Unclassified 0 0 0
```

```
=====
USER PID PPID PRI NI DATA TEXT PAGE TIME CPU% I/O OTH COMMAND
root 1 0 108 20 197 9 180 0:24 0.0 0 0 init
root 1032 0 16 41 3 3374 3 0:00 0.0 0 0 lrud
root 1290 0 60 41 4 3374 4 0:02 0.0 0 0 xmgc
root 1548 0 36 41 4 3374 4 0:26 0.0 0 0 netm
root 1806 0 37 41 16 3374 16 13:25 0.0 0 0 gil
root 2064 0 16 41 4 3374 4 0:04 0.0 0 0 wlmsched
root 2698 1 108 20 14 2 14 0:00 0.0 0 0 shlap
root 3144 1 108 20 40 1 36 5:19 0.0 0 0 syncd
root 3362 0 108 20 4 3374 4 0:00 0.0 0 0 lvmbb
root 3666 1 108 20 135 23 123 0:00 0.0 0 0 errdemon
root 3982 0 108 20 4 3374 4 0:01 0.0 0 0 rtcmd
```

Flags

-cecdisp Displays the cross-partition panel.

-cpus	Specifies the number of hot CPUs to be monitored. This is also the maximum number of CPUs displayed when enough room is available on the screen. If this number exceeds the number of CPUs available, only the installed CPUs will be monitored and displayed. If this argument is omitted, a default of 2 is assumed. If a value of 0 (zero) is specified, no CPU information is monitored.
-disks	Specifies the number of disks to be monitored. This is also the maximum number of disks displayed when enough room is available on the screen. When this number exceeds the number of disks installed, only the installed disks will be monitored and displayed. If this argument is omitted, a default of 2 is assumed. If a value of 0 (zero) is specified, no disk information is monitored.
-interval	Sets the monitoring interval in seconds. The default is 2 seconds.
-nets	Specifies the number of hot network interfaces to be monitored. This is also the maximum number of network interfaces displayed when enough room is available on the screen. When this number exceeds the number of network interfaces installed, only the installed network interfaces will be monitored and displayed. If this argument is omitted, a default of 2 is assumed. If a value of 0 (zero) is specified, no network information is monitored.
-procsdisp	Displays the full-screen process display. This display shows a list of the busiest processes, similar to the process subsection on the default display, only with more columns showing more metrics per process. This list can be sorted by any column.
-procs	Specifies the number of hot processes to be monitored. This is also the maximum number of processes shown when enough room is available on the screen. If this argument is omitted, a default of 20 is assumed. If a value of 0 is specified, no process information will be monitored. Retrieval of process information constitutes the majority of the topas overhead. If process information is not required, always use this option to specify that you do not want process information.
-wlmdisp	Displays the full-screen WLM class display, which is a split display. The top part of the display shows a list of hot WLM classes, similar to the WLM classes subsection on the default display, but with enough space available to display the full class names. This list can be sorted on any column. The bottom part of the display shows a list of busiest processes, similar to the full screen process display, but only displays processes belonging to one WLM class (selected with the f key).
-wlms	Specifies the number of hot WorkLoad Management (WLM) classes to be monitored. This is also the maximum number of WLM classes displayed when enough room is available on the screen. If this number exceeds the number of WLM classes installed, only the installed WLM classes will be monitored and displayed. If this argument is omitted, a default of 2 is assumed. If a value of 0 (zero) is specified, no WLM class information is monitored.

Subcommands

While **topas** is running, it accepts one-character subcommands. Each time the monitoring interval elapses, the program checks for one of the following subcommands and responds to the action requested.

a	The a key shows all of the variable subsections being monitored (CPU, network, disk, WLM, and process). Pressing the a key always returns the topas command to the initial main display.
c	The c key toggles the CPU subsection between the cumulative report, off, and a list of the busiest CPUs. The number of busiest CPUs displayed will depend upon the space available on the screen.
d	The d key toggles the disk subsection between a list of busiest disks, off, and the report on the total disk activity of the system. The number of busiest disks displayed will depend upon the space available on the screen.
h	Show the help screen.

n	The n key toggles the network interfaces subsection between a list of busiest interfaces, off, and the report on the total network activity of the system. The number of busiest interfaces displayed will depend upon the space available on the screen.
w	The w key toggles the WorkLoad Management (WLM) classes subsection on and off. The number of busiest WLM classes displayed will depend upon the space available on the screen.
p	The p key toggles the hot processes subsection on and off. The number of busiest processes displayed will depend upon the space available on the screen.
P	The uppercase P key replaces the default display with the full-screen process display. This display provides more detailed information about processes running on the system than the process section of the main display. When the P key is pressed again, it toggles back to the default main display.
W	The uppercase W key replaces the default display with the full-screen WLM class display. This display provides more detailed information about WLM classes and processes assigned to classes. When the W key is pressed again, it toggles back to the default main display.
L	The uppercase L key replaces the current display with the logical partition display; Micro-Partitioning™ and SMT metrics similar to what lparstat and mpstat provide are displayed.
f	Moving the cursor over a WLM class and pressing the f key displays the list of top processes in the class at the bottom of the WLM screen. This key is valid only when topas is in the full-screen WLM display (by using the W key or the -wlms flag).
q	Quit the program.
r	Refresh the display.
Arrow and Tab keys	Subsections from the main display such as the CPU, Network, Disk, WLM Classes, and the full-screen WLM and Process displays can be sorted by different criteria. Positioning the cursor over a column activates sorting on that column. The entries are always sorted from highest to lowest value. The cursor can be moved by using the Tab key or the arrow keys. Sorting is only valid for 128 disks and 16 network adapters.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

1. To display up to twenty “hot” disks every five seconds and omit network interface, WLM classes, and process information, type:

```
topas -interval 5 -nets 0 -procs 0 -wlms 0
```
2. To display the five most active processes and up to twenty most active WLM classes (which is the default when omitting the **-w** flag) but no network or disk information, type:

```
topas -procs 5 -nets 0 -disks 0
```
3. To run the program with default options, type:

```
topas
```
4. To go directly to the process display, type:

```
topas -procdisp
```
5. To go directly to the WLM classes display, type:

```
topas -wlmdisp
```

traceroute command

Purpose

Prints the route that IP packets take to a network host.

Syntax

```
traceroute [ -hops Hops ] [ -num ] [ -port Port ] [ -src Address ] Host [ PacketSize ]
```

Description

The **traceroute** command attempts to trace the route an IP packet follows to an Internet host by launching UDP probe packets with a small maximum time-to-live (*Hops* parameter), then listening for an **ICMP TIME_EXCEEDED** response from gateways along the way. Probes are started with a Hops value of one hop, which is increased one hop at a time until an **ICMP PORT_UNREACHABLE** message is returned. The **ICMP PORT_UNREACHABLE** message indicates either that the host has been located or the command has reached the maximum number of hops allowed for the trace.

The **traceroute** command sends three probes at each Hops setting to record the following:

- Hops value
- Address of the gateway
- Round-trip time of each successful probe

If the probe answers come from different gateways, the command prints the address of each responding system. If there is no response from a probe within a 3-second time-out interval, an * (asterisk) is printed for that probe.

Note: The **traceroute** command is intended for use in network testing, measurement, and management. It should be used primarily for manual fault isolation. Because of the load it imposes on the network, the traceroute command should not be used during normal operations or from automated scripts.

The **traceroute** command prints an ! (exclamation mark) after the round-trip time if the Hops value is one hop or less. A maximum time-to-live value of one hop or less generally indicates an incompatibility in the way ICMP replies are handled by different network software. The incompatibility can usually be resolved by doubling the last Hops value used and trying again.

Other possible annotations after the round-trip notation are as follows:

```
!H      Host unreachable
!N      Network unreachable
!P      Protocol unreachable
!S      Source route failed
!F      Fragmentation needed
```

If the majority of probes result in an error, the **traceroute** command exits.

The only mandatory parameter for the **traceroute** command is the destination host name or IP number. The **traceroute** command will determine the length of the probe packet based on the Maximum Transmission Unit (MTU) of the outgoing interface. The UDP probe packets are set to an unlikely value so as to prevent processing by the destination host.

Flags

-hops <i>Max_ttl</i>	Sets the maximum time-to-live (maximum number of hops) used in outgoing probe packets. The default is 30 hops (the same default used for TCP connections).
-num	Prints hop addresses numerically rather than symbolically and numerically. This flag saves a name-server address-to-name lookup for each gateway found on the path.

- port** *Port* Sets the base UDP port number used in probes. The default is 33434. The **traceroute** command depends on an open UDP port range of base to base + nhops - 1 at the destination host. If a UDP port is not available, this option can be used to pick an unused port range.
- src** *SRC_Addr* Uses the next IP address in numerical form as the source address in outgoing probe packets. On hosts with more than one IP address, the **-src** flag can be used to force the source address to be something other than the IP address of the interface on which the probe packet is sent. If the next IP address is not one of the machine's interface addresses, an error is returned and nothing is sent.

Parameters

- Host* Specifies the destination host, either by host name or IP number. This parameter is required.
- PacketSize* Specifies the probe datagram length. The default packet size is determined by the **traceroute** command based on the MTU of the outgoing interface.

Exit Status

See "Exit status for Virtual I/O Server commands" on page 1.

Examples

- To print the route to host **nis.nsf.net**, type:

```
traceroute nis.nsf.net
```

The output will look similar to the following:

```
traceroute to rotterdam (35.1.1.48), 30 hops max, 56 byte packet
 1 helios.ee.lbl.gov (128.3.112.1) 19 ms 19 ms 0 ms
 2 lilac-dmc.Berkeley.EDU (128.32.216.1) 39 ms 39 ms 19 ms
 3 lilac-dmc.Berkeley.EDU (128.32.216.1) 39 ms 39 ms 19 ms
 4 ccngw-ner-cc.Berkeley.EDU (128.32.136.23) 39 ms 40 ms 39 ms
 5 ccn-nerif22.Berkeley.EDU (128.32.168.22) 39 ms 39 ms 39 ms
 6 128.32.197.4 (128.32.197.4) 40 ms 59 ms 59 ms
 7 131.119.2.5 (131.119.2.5) 59 ms 59 ms 59 ms
 8 129.140.70.13 (129.140.70.13) 99 ms 99 ms 80 ms
 9 129.140.71.6 (129.140.71.6) 139 ms 239 ms 319 ms
10 129.140.81.7 (129.140.81.7) 220 ms 199 ms 199 ms
11 nic.merit.edu (35.1.1.48) 239 ms 239 ms 239 ms
```

Related Information

The ping command and the optimizenet command.

uname command

Purpose

Writes to standard output the name of the operating system that you are using

Syntax

```
uname [ -a ] [-f ] [-F] [-l] [-L] [-m] [-M] [-n] [-p] [-r] [-s] [-u] [-x]
```

Description

The machine ID number contains 12 characters in the following digit format: *xyyyyyyyymmss*. The *xx* positions indicate the system and is always 00. The *yyyyyy* positions contain the unique ID number for the entire system. The *mm* position represents the model ID. The *ss* position is the submodel number and is always 00. The model ID describes the ID of the CPU Planar, not the model of the System as a whole. You can use the **uname -m** command sometimes to determine which model you are using. The following list is not complete. Refer to hardware vendor supplied documentation for values in the range E0 - FF. Also note that not all machine types have a machine ID. Many new machines share a common machine ID of 4C.

Flags

-a	Displays all information specified with the -m , -n , -r , -s , and -v flags. Cannot be used with the -x flag. If the -x flag is specified with the -a flag, the -x flag overrides it.
-F	Displays a system identification string comprised of hexadecimal characters. This identification string is the same for all partitions on a particular system.
-f	Similar to the F flag, except that the partition number is also used in the calculation of this string. The resulting identification string is unique for each partition on a particular system.
-l	Displays the LAN network number.
-L	Displays LPAR number and LPAR name. If LPAR does not exist, -1 is displayed for LPAR number and NULL for LPAR name.
-m	Displays the machine ID number of the hardware running the system. Note: The -m flag cannot be used to generate a unique machine identifier for partitions in an LPAR environment.
-M	Displays the system model name. If the model name attribute does not exist, a null string is displayed.
-n	Displays the name of the node. This may be a name the system is known by to a UUCP communications network.
-p	Displays the architecture of the system processor.
-r	Displays the release number of the operating system.
-s	Displays the system name. This flag is on by default.
-u	Displays the system ID number. If this attribute is not defined, the output is the same as the output displayed by uname -m .
-x	Displays the information specified with the -a flag as well as the LAN network number, as specified by the -l flag.

Exit Status

The following exit values are returned:

0	Successful completion.
>0	An error occurred.

Examples

1. To display the complete system name and version banner, enter:

```
uname -a
AIX vios_bat 3 5 00CD1B0E4C00
```

2. To display the operating system name, enter:

```
uname
AIX
```

Virtual I/O Server **uname** command calls the AIX **uname** command. The flags are the same except that **-S**, **-T**, and **-v** are not allowed.

Related Information

uname subroutine

unloadopt command

Purpose

Remove a virtual optical media disk from a virtual optical device.

Syntax

```
unloadopt [ -release ] -vtd VirtualTargetDevice
```

Description

The **unloadopt** command removes a virtual optical disk from the specified virtual optical device (*VirtualTargetDevice*). After the command completes the specified optical device will contain no media.

Flags

-release	Forces the virtual optical device to be unlocked even if the client has a reserve on the device.
-vtd <i>VirtualTargetDevice</i>	The name of the virtual target device.

Examples

To unload the virtual optical disk loaded into the virtual optical device vopt1, type the following command:

```
unloadopt -vtd vopt1
```

unmirrorios command

Purpose

Removes the mirrors that exist on the rootvg volume group.

Syntax

```
unmirrorios [ PhysicalVolume ...]
```

Description

The **unmirrorios** command unmirrors all the logical volumes detected on the rootvg volume group. By default, **unmirrorios** will pick the set of mirrors to remove from a mirrored volume group. To control which drives no longer are to contain mirrors, you must include the list of disks in the input parameters, *PhysicalVolume*.

At the completion of this command, Quorum will be disabled until the system is rebooted.

Note: If LVM has not recognized that a disk has failed, it is possible that it will remove a different mirror. If you know that a disk has failed, and LVM does not show those disks as missing, you should specify the failed disks on the command line or you should use the **reducevg** command to remove the disk.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Security

This command can only be executed by the prime administrator.

Related Information

The activatevg command, the chvg command, the deactivatevg command, the exportvg command, the importvg command, the lsvg command, the mkvg command, the syncvg command, and the mirrorios command.

unmount command

Purpose

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

Syntax

```
unmount { Directory | File | FileSystem }
```

Description

The **unmount** command unmounts a previously mounted directory, file, or file system. Processing on the file system, directory, or file completes and it is unmounted.

To unmount local mounts you can specify the device, directory, file, or file system on which it is mounted.

Exit Status

See “Exit status for Virtual I/O Server commands” on page 1.

Examples

1. To unmount files and directories, type the following command:

```
unmount /home/user/test
```

This unmounts the filesystem mounted at **/home/user/test**.

Related Information

The mount command.

IVM update_install_setup command

Purpose

Saves late binding installation information into the system plan.

Syntax

```
update_install_setup -f plan name [-m managed system name ] -id partition id | -p partition name -G gateway  
-C client IP -sm subnet mask [-s speed -d duplex] -n resource name [-help]
```

```
update_install_setup -l -f plan name [-m managed system name ] [-id partition id | -p partition name] [-F  
attribute names] [-help]
```

Description

The **update_install_setup** command saves the late binding installation parameters, such as the client IP and subnet mask, into the system plan existing on this Integrated Virtualization Manager (IVM). It also lists the information already saved in the system plan.

Flags

-f <i>plan name</i>	Specifies the name of the system plan file that is to be updated.
-m <i>managed system</i>	Specifies the name of the managed system.
-p <i>partition name</i>	Specifies the name of the logical partition for which the network information needs to be updated into the system plan file.
-id <i>partition id</i>	Specifies the partition ID of the logical partition for which the network information needs to be updated into the system plan file.
-G <i>gateway</i>	Specifies the gateway IP address.
-C <i>client IP</i>	Specifies the client IP address.
-sm <i>subnet mask</i>	Specifies the subnet mask of the client interface.
-s <i>speed</i>	Specifies the speed of the network adapter.
-d <i>duplex</i>	Specifies the network adapter duplex.
-n <i>resource name</i>	Specifies the name of the system plan resource.
-help	Displays the help text for this command and exit.
-l	Lists the installation parameters.
-F <i>attribute names</i>	A delimiter-separated list of attribute names for the desired attribute values to be displayed for each resource. If no attribute names are specified, values for all of the attributes for the resource are displayed.
	When this option is specified, only attribute values are displayed. No attribute names are displayed.
	The attribute values displayed are separated by the delimiter that was specified with this option. This option is useful when you want only the attribute values to be displayed, or when you want the values of only selected attributes to be displayed.
-header	Prints a header of attribute names when the -F flag is also specified.

Exit Status

The following exit values are returned:

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

Examples

1. To save the late binding information into a system plan named `test.sysplan` for logical partition `client1` with the client IP address `9.3.126.77`, subnet mask `255.255.255.0`, gateway `9.3.126.1`, and resource name `AIX53`, type:

```
update_install_setup -f test.sysplan -p client1 -C 9.3.126.77  
-sm 255.255.255.0 -G 9.3.126.1 -n AIX53
```

2. List the late binding installation information stored in a system plan named `test.sysplan` on the IVM:

```
update_install_setup -f test.sysplan -l
```

Related Information

The `mksysplan`, `deploysysplan`, and `lssysplan` commands.

updateios command

Purpose

Updates the Virtual I/O Server to latest maintenance level.

Syntax

```
updateios -dev Media [-f] [ -install ] [ -accept ]
```

```
updateios -commit | -reject [ -f ]
```

```
updateios -cleanup
```

```
updateios -remove { -file RemoveListFile | RemoveList }
```

Description

The **updateios** command is used to install fixes, or to update the Virtual I/O Server to the latest maintenance level. Before installing a fix or updating the maintenance level, the **updateios** command runs a preview installation and displays the results. You are then prompted to continue or exit. If the preview fails for any reason, do not install the update.

Flags

-accept	Specifies that you agree to required software license agreements for software to be installed.
-cleanup	Specify the clean up flag to remove all incomplete pieces of the previous installation. Perform cleanup processing whenever any software product or update is after an interrupted installation or update is in a state of either applying or committing. You can run this flag manually, as needed.
-commit	The -commit flag will commit all uncommitted updates to the Virtual I/O Server.
-dev <i>Media</i>	Specifies the device or directory containing the images to install.
-f	Forces all uncommitted updates to be committed prior to applying the new updates. When combined with the -dev flag, this flag commits all updates prior to applying any new ones. When combined with the -reject flag, this flag rejects all uncommitted updates without prompting you for confirmation.
-file <i>file</i>	Specifies the file containing a list of entries to uninstall.
-install	The -install flag is used to install new and supported file sets onto the Virtual I/O Server.
	Attention: The log file, install.log in the user's home directory, will be overwritten with a list of all file sets that were installed.
-reject	The -reject flag will reject all uncommitted updates to the Virtual I/O Server. If dependent software is in a committed state for a specific update, then that update cannot be rejected.
-remove	If the -remove flag is specified, the listed file sets will be removed from the system. The file sets to be removed must be listed on the command line or in the <code>RemoveListFile</code> file.

Exit Status

The following exit values are returned:

19	All uncommitted updates must be committed
20	There are no uncommitted updates

Examples

1. To update the Virtual I/O Server to the latest level, where the updates are located on the mounted file system `/home/padmin/update`, type the following command:
`updateios -dev /home/padmin/update`
2. To update the Virtual I/O Server to the latest level, when previous levels are not committed, type the following command:
`updateios -f -dev /home/padmin/update`
3. To reject installed updates, type the following command:
`updateios -reject`
4. To cleanup partially installed updates, type the following command:
`updateios -cleanup`
5. To commit the installed updates, type the following command:
`updateios -commit`

Related Information

The `lssw` command, the `ioslevel` command, the `remote_management` command, the `oem_setup_env` command, and the `oem_platform_level` command.

vasistat command

Purpose

Shows VASI device driver and device statistics.

Syntax

```
vasistat [ -all ] [ -reset ] [ -debug ] Device_Name
```

Description

The `vasistat` command displays the statistics gathered by the specified VASI device driver. The user can optionally specify that the device-specific statistics be displayed in addition to the device generic statistics. This will display statistics for all the operations, such as migrations, in that VASI device. If no flags are specified, only the device generic statistics are displayed.

Flags

<code>-all</code>	Displays all the statistics, including the device-specific statistics.
<code>-reset</code>	Resets all the statistics back to their initial values.
<code>-debug</code>	Toggles internal debug tracing in device driver

Parameters

<code>-all</code>	The name of the VASI device, for example, <code>vasi0</code> .
-------------------	--

Exit Status

The statistic fields displayed in the output of the `vasistat` command and their descriptions are as follows:

Device Type

Displays the description of the adapter type.

Elapsed Time

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

Transmit statistics fields

Table 2. Transmit statistics fields and descriptions.

Statistic	Description
Packets	The number of packets transmitted successfully to PHYP by the device.
Bytes	The number of bytes transmitted successfully to PHYP by the device.
Transmit Errors	The number of output errors encountered on this device. This is a counter for unsuccessful transmissions due to errors returned by PHYP.
Bad Packets	The number of outgoing packets that could not be sent because they were malformed (for example, packets exceeding the VASI MTU size).
No Buffers	The number of times a packet could not be sent to PHYP because there were no transmit buffers available for sending.
Interrupts	The number of transmit interrupts (for example, attempts at transmitting a packet to PHYP).

Receive Statistics Fields

Table 3. Receive statistics fields and descriptions.

Statistic	Description
Packets	The number of packets received successfully from PHYP by the device.
Bytes	The number of bytes received successfully from PHYP by the device.
Receive Errors	The number of receive errors encountered on this device (for example, bad packets).
Bad Packets	The number of incoming packets that could not be processed because they were malformed (for example, VASI could not figure out to which operation the data belonged to).
No Buffers	The number of times VASI tried to allocate a system buffer but there was no memory to do so. This is a soft error since the data will still be delivered in the original buffer.
Interrupts	The number of receive interrupts (for example, number of CRQ entries received by VASI).
System Buffers	The number of system buffers allocated when the receive buffer pools are low on buffers.

Miscellaneous Statistics Fields

Table 4. Miscellaneous statistics fields and descriptions.

Statistic	Description
Interrupt Processing Exceeded	The number of times this device attempted to process more packets on the interrupt context than the maximum allowed.

Table 4. Miscellaneous statistics fields and descriptions. (continued)

Statistic	Description
Offlevel Interrupt Scheduled	The number of times an offlevel interrupt was scheduled on this device, in order to handle packets that could not be handled on the interrupt context.
Maximum Operations	The maximum number of operations (for example, migrations) this device can handle simultaneously.
Maximum Receive Pools	The maximum number of different receive pool sizes this device can handle.
Active Operations	The number of operations (for example, migrations) that are currently active on this device.

Operation-specific Statistics Fields

These statistics are displayed for each operation, either active or inactive. Inactive applications will have the text `INACTIVE` in the title header, and the text `INVALID STREAM ID` will appear in the Stream ID field. None of these statistics will be zeroed out by the `-reset` flag.

Table 5. Operation-specific statistics fields and descriptions.

Statistic	Description
Operation Type	The type of this operation (for example, migration) as well as any pertinent information (for example, whether it is the source or target of the migration).
Stream ID	The unique number that identifies this operation; the text <code>INVALID STREAM ID</code> denotes that this operation is no longer active.
TOP/BOTTOM	The opaque identifiers used by the VASI device and PHYP to refer to this operation.
Elapsed Time	Displays the real time period which has elapsed since the operation was started. This time will stop when the operation is terminated, so it can be used to measure how long the operation took to complete.
Flags	Denotes values that may be used to describe this operation: <ol style="list-style-type: none"> 1. <code>RUNNABLE</code> - this operation has been initialized and is ready to receive CRQ commands 2. <code>TERMINATED</code> - this operation has been terminated internally due to some error within VASI
Operation State	State the operation is in with regards to its setup: <ol style="list-style-type: none"> 1. <code>Not Operational</code> - no memory has been allocated for this operation 2. <code>Work Queues Allocated</code> - the work queues (where incoming CRQ responses are queued) have been allocated 3. <code>PHYP Registered</code> - VASI has established a connection with PHYP on behalf of this operation 4. <code>Pools Partially Allocated</code> - some receive buffer pools have been allocated 5. <code>Pools Allocated</code> - all receive buffer pools have been allocated 6. <code>Buffers Partially Registered</code> - some receive buffers have been registered with PHYP 7. <code>Operational</code> - all setup is complete

Table 5. Operation-specific statistics fields and descriptions. (continued)

Statistic	Description
Stream State	State the stream (for example, migration) is in; these values will vary depending on the operation in question. For migration, these states are as follows: <ol style="list-style-type: none"> 1. Invalid - the migration has not been started 2. Enabled - the migration has been started 3. Aborted - the migration has aborted (may be due to internal VASI error, user-initiated abort, Mover-initiated abort, or PHYP-initiated abort) 4. Suspending - the suspend trigger percentage has been reached and the migrating partition may be suspended 5. Suspended - the migrating partition has been suspended at the source 6. Resumed - the migrating partition has been resumed at the target 7. Completed - the migration has completed successfully
Total Bytes to Transfer	The estimated total number of bytes to transfer for this operation
Bytes Left to Transfer	The estimated number of bytes that are left to transfer for this operation

Receive Buffer Usage Statistics Fields

These statistics are displayed only for operations that are currently active. For every receive buffer pool the following information is shown in tabular form:

Table 6. Receive buffer usage statistics fields and descriptions.

Statistic	Description
Size	The size in bytes of the packets of the buffer pool.
Reg	The number of buffers from the buffer pool that are currently registered with PHYP.
Alloc	The number of buffers that have been allocated for the buffer pool.
Max	The maximum number of buffers that could ever be allocated to this buffer pool.
LowReg	The lowest number of buffers from this pool that have ever been registered with PHYP.

Transmit Buffer Usage Statistics Fields

These statistics are displayed only for operations that are currently active.

Table 7. Transmit buffer usage statistics fields and descriptions.

Statistic	Description
Number of Buffers	The number of buffers that have been allocated for the transmit buffer pool.
Buffer Size	The size in bytes of the packets of the transmit buffer pool.
Mapped	The number of buffers in the transmit buffer pool that have been DMA-mapped.
Next Available Buffer	The index of the next buffer to be used in the transmit buffer pool.
In Use	The number of buffers in the transmit buffer pool that are currently being used.
Maximum Used	The maximum number of buffers in the transmit buffer pool that have ever been used concurrently.

Examples

1. To display the device generic statistics for vasi0, type:

```
vasistat vasi0
```

This produces output similar to the following:

VASI STATISTICS (vasi0) :

Device Type: Virtual Asynchronous Services Interface (VASI)

Elapsed Time: 0 days 0 hours 10 minutes 38 seconds

Transmit Statistics:	Receive Statistics:
-----	-----
Packets: 2	Packets: 179098
Bytes: 224	Bytes: 753605020
Transmit Errors: 0	Receive Errors: 0
Bad Packets: 0	Bad Packets: 0
No Buffers: 0	No Buffers: 0
Interrupts: 2	Interrupts: 214876
	System Buffers: 0

Interrupt Processing Exceeded: 0

Offlevel Interrupt Scheduled: 0

Driver Flags: Up Running 64BitSupport

Maximum Operations: 4

Maximum Receive Pools: 3

Active Operations: 1

2. To display the VASI device generic statistics and the statistics for all the operations for vasi0, type:

vasistat -all vasi0

This produces results similar to the following:

VASI STATISTICS (vasi0) :

Device Type: Virtual Asynchronous Services Interface (VASI)

Elapsed Time: 0 days 0 hours 10 minutes 38 seconds

Transmit Statistics:	Receive Statistics:
-----	-----
Packets: 2	Packets: 179098
Bytes: 224	Bytes: 753605020
Transmit Errors: 0	Receive Errors: 0
Bad Packets: 0	Bad Packets: 0

No Buffers: 0 No Buffers: 0
Interrupts: 2 Interrupts: 214876
 System Buffers: 0

Interrupt Processing Exceeded: 0
Offlevel Interrupt Scheduled: 0

Driver Flags: Up Running 64BitSupport

Maximum Operations: 4
Maximum Receive Pools: 3
Active Operations: 1

Statistics for each operation:
=====

Operation #0 (ACTIVE):

Operation Type: Migration (Source)
Stream ID: 0000000000000001
TOP/BOTTOM: 00000000/00040000
Elapsed Time: 0 days 0 hours 0 minutes 11 seconds
Flags: <RUNNABLE>
Operation State: Operational
Stream State: Enabled
Total Bytes to Transfer: 546832384
Bytes Left to Transfer: 360833024

Transmit Statistics:	Receive Statistics:
-----	-----
Packets: 1	Packets: 45415
Bytes: 112	Bytes: 191086638
Transmit Errors: 0	Receive Errors: 0
Bad Packets: 0	Bad Packets: 0

No Buffers: 0 No Buffers: 0
Interrupts: 1 Interrupts: 0
 System Buffers: 0

Receive Buffer Usage:

	Size	Reg	Alloc	Max	LowReg
Pool #0	8192	198	256	256	193
Pool #1	2048	2048	2048	2048	2044

Transmit Buffer Usage:

Number of Buffers: 64
Buffer Size: 16384 bytes
Mapped: 64
Next Available Buffer: 0
In Use: 0
Maximum Used: 1

Operation #1 (INACTIVE):

Operation Type: Unknown (Unknown)
Stream ID: ** INVALID STREAM ID **
TOP/BOTTOM: 00000000/00000000
Elapsed Time: 0 days 0 hours 0 minutes 0 seconds
Flags:
Operation State: Not Operational
Stream State: Unknown
Total Bytes to Transfer: 0
Bytes Left to Transfer: 0

Transmit Statistics: Receive Statistics:

```

-----
Packets: 0          Packets: 0
Bytes: 0            Bytes: 0
Transmit Errors: 0  Receive Errors: 0
Bad Packets: 0     Bad Packets: 0
No Buffers: 0      No Buffers: 0
Interrupts: 0      Interrupts: 0

                System Buffers: 0

```

Operation #2 (INACTIVE):

```

-----
Operation Type: Unknown (Unknown)
Stream ID: ** INVALID STREAM ID **
TOP/BOTTOM: 00000000/00000000
Elapsed Time: 0 days 0 hours 0 minutes 0 seconds
Flags:
Operation State: Not Operational
Stream State: Unknown
Total Bytes to Transfer: 0
Bytes Left to Transfer: 0

```

```

Transmit Statistics:      Receive Statistics:
-----
Packets: 0                Packets: 0
Bytes: 0                   Bytes: 0
Transmit Errors: 0        Receive Errors: 0
Bad Packets: 0            Bad Packets: 0
No Buffers: 0             No Buffers: 0
Interrupts: 0             Interrupts: 0

                System Buffers: 0

```

Operation #3 (INACTIVE):

```

-----
Operation Type: Unknown (Unknown)

```

```
Stream ID: ** INVALID STREAM ID **
TOP/BOTTOM: 00000000/00000000
Elapsed Time: 0 days 0 hours 0 minutes 0 seconds
Flags:
Operation State: Not Operational
Stream State: Unknown
Total Bytes to Transfer: 0
Bytes Left to Transfer: 0
```

```
Transmit Statistics:      Receive Statistics:
-----
Packets: 0                Packets: 0
Bytes: 0                   Bytes: 0
Transmit Errors: 0        Receive Errors: 0
Bad Packets: 0            Bad Packets: 0
No Buffers: 0             No Buffers: 0
Interrupts: 0             Interrupts: 0
                          System Buffers: 0
```

vfcmap command

Purpose

Maps the virtual fibre channel adapter to the physical fibre channel port.

Syntax

```
vfcmap -vadapter virtual fibre channel adapter -fcp fibre channel port name
```

Description

The **vfcmap** command maps or unmaps the virtual fibre channel adapter to the physical fibre channel port.

Flags

-vadapter *virtual fibre channel adapter*
-fcp *fibre channel port name*

Specifies the virtual server adapter.

Specifies the physical fibre channel port.

Note: If no parameter is specified with this flag, the command unmaps the virtual fibre channel adapter from the physical fibre channel port.

Exit Status

The following exit values are returned:

0	Successful completion.
>0	Invalid flag, argument, or command failure.

Examples

1. To map the virtual fibre channel, **vfchost7**, to the physical fibre channel port, **fcs0**, type:

```
vfcmmap -vadapter vfchost7 -fcp fcs0
```

The system displays a message similar to the following:

```
vfchost0 changed
```

2. To unmap the virtual fibre channel, **vfchost7**, from any physical fibre channel port, type:

```
vfcmmap -vadapter vfchost7 -fcp
```

The system displays a message similar to the following:

```
vfchost0 changed
```

Related Information

The **lsmmap** and **lsmports** commands.

vi command

Purpose

Edits files with a full-screen display.

Syntax

```
{ vi | vedit } [ -l ] [ -R ] [ -tTag ] [ -v ] [ -wNumber ] [ -yNumber ] [ -r [ File ] ] [ { + | -c } { Subcommand } ] [ File ... ]
```

Description

The **vi** command starts a full-screen editor based on the underlying ex editor. Therefore, ex subcommands can be used within the vi editor. The **vedit** command starts a version of the vi editor intended for beginners. In the vedit editor, the **report** option is set to 1, the **showmode** option is set, and the **novice** option is set, making it a line editor.

You start the vi editor by specifying the name of the file or files to be edited. If you supply more than one *File* parameter on the command line, the vi editor edits each file in the specified order. The vi editor on an existing file displays the name of the file, the number of lines, and the number of characters at the bottom of the screen. In case of multibyte locales the number of characters need to be interpreted as the number of bytes.

Since the vi editor is a full-screen editor, you can edit text on a screen-by-screen basis. The vi editor makes a copy of the file you are editing in an edit buffer, and the contents of the file are not changed until you save the changes. The position of the cursor on the display screen indicates its position within the file, and the subcommands affect the file at the cursor position.

vi Editor Limitations

The following list provides the maximum limits of the vi editor. These counts assume single-byte characters.

- 8192 characters per line
- 256 characters per global command list
- 128 characters in the previously inserted and deleted text
- 128 characters in a shell escape command
- 128 characters in a string-valued option
- 30 characters in a tag name
- 128 map macros with 2048 characters total
- 1,048,560 lines of 8192 characters per line silently enforced
- The macro name and the macro text are limited to 100 characters.

Note: Running the vi editor on a file larger than 64 MB may cause the following error message to display:

```
0602-103 file too large to place in /tmp
```

vi Editing Modes

The vi editor operates in the following modes:

command mode	When you start the vi editor, it is in command mode. You can enter any subcommand except those designated for use only in the text input mode. The vi editor returns to command mode when subcommands and other modes end. Press the Esc key to cancel a subcommand.
text-input mode	You use the vi editor in this mode to add text. Enter text input mode with any of the following subcommands: the a subcommand, A subcommand, i subcommand, I subcommand, o subcommand, O subcommand, cx subcommands (where the <i>x</i> represents the scope of the subcommand), C subcommand, s subcommand, S subcommand, and R subcommand. After entering one of these subcommands, you can enter text into the editing buffer. To return to command mode, press the Esc key for normal exit or press Interrupt (the Ctrl-C key sequence) to end abnormally.
last-line mode	Subcommands with the prefix : (colon), / (slash), ? (question mark), ! (exclamation point), or !! (two exclamation points) read input on a line displayed at the bottom of the screen. When you enter the initial character, the vi editor places the cursor at the bottom of the screen, where you enter the remaining characters of the command. Press the Enter key to run the subcommand, or press Interrupt (the Ctrl-C key sequence) to cancel it. When the !! prefix is used, the cursor moves only after both exclamation points are entered. When you use the : prefix to enter the last-line mode, the vi editor gives special meaning to the following characters when they are used before commands that specify counts: % All lines regardless of cursor position \$ Last line . Current line

Customizing the vi Editor

You can customize the vi editor by:

- Setting vi editor options
- Defining macros

- Mapping keys
- Setting abbreviations

Setting vi Editor Options

The following list describes the vi editor options you can change with the **set** command. The default setting for these options is **off**. If you turn on one of these toggle options, you can turn it off again by entering the word **no** before the option. If you want to discontinue the **autowrite** vi option, enter **noaw**, where **no** turns off the option and **aw** specifies the **autowrite** option.

Note: Do not include parentheses when entering vi options.

vi Option (Abbreviation)	Description
autoindent (ai)	Indents automatically in text input mode to the indentation of the previous line by using the spacing between tab stops specified by the shiftwidth option. The default is noai . To back the cursor up to the previous tab stop, press the Ctrl-D key sequence. This option is not in effect for global commands.
autoprin (ap)	Prints the current line after any command that changes the editing buffer. The default is ap . This option applies only to the last command in a sequence of commands on a single line and is not in effect for global commands.
autowrite (aw)	Writes the editing buffer to the file automatically before the :n subcommand, the :ta subcommand, the Ctrl-A, Ctrl -], and Ctrl -T key sequences, and the ! subcommand if the editing buffer changed since the last write subcommand. The default is noaw .
backtags (bt)	Allows the Ctrl-T subcommand to return the file editing position to the location where the previous Ctrl-] subcommand was issued. If nobacktags is set, then Ctrl-T is the same as Ctrl-]. The default is backtags .
beautifying text (bf)	Prevents the user from entering control characters in the editing buffer during text entry (except for tab, new-line, and form-feed indicators). The default is nofb . This option applies to command input.
closepunct (cp=)	Handles a list of closing punctuation, especially when wrapping text (wraptyp option). Precedes multicharacter punctuation with the number of characters; for example, cp=3. ;) } . The vi command does not split closing punctuation when wrapping.
directory (dir=)	Displays the directory that contains the editing buffer. The default is dir = /var/tmp .
edcompatible (ed)	Retains g (global) and c (confirm) subcommand suffixes during multiple substitutions and causes the r (read) suffix to work like the r subcommand. The default is noed .
exrc (exrc)	If not set, ignores any .exrc file in the current directory during initialization, unless the current directory is that named by the HOME environment variable. The default is noexrc .
hardtabs (ht=)	Tells the vi editor the distance between the hardware tab stops on your display screen. (This option must match the tab setting of the underlying terminal or terminal emulator.) The default is ht=8 .
ignorecase (ic)	Ignores distinction between uppercase and lowercase while searching for regular expressions. The default is noic .
linelimit (ll=)	Sets the maximum number of lines, as per the -y command-line option. This option only is effective if used with the .exrc file or the EXINIT environment variable.
lisp (lisp)	Removes the special meaning of (), { }, [], and] and enables the = (formatted print) operator for s-expressions, so you can edit list processing (LISP) programs. The default is noisp .

vi Option (Abbreviation)	Description
list (list)	Displays text with tabs (^I) and the marked end of lines (\$). The default is nolist .
magic (magic)	Treats the . (period), [(left bracket), and * (asterisk) characters as special characters when searching for a pattern. In off mode, only the () (parentheses) and \$ (dollar sign) retain special meanings. However, you can evoke special meaning in other characters by preceding them with a \ (backslash). The default is magic .
mesg (mesg)	Turns on write permission to the terminal if set while in visual mode. This option only is effective if used with the .exrc file or the EXINIT environment variable. The default is on .
modeline (modeline)	Runs a vi editor command line if found in the first five or the last five lines of the file. A vi editor command line can be anywhere in a line. For the vi editor to recognize a command line, the line must contain a space or a tab followed by the ex: or vi: string. The command line is ended by a second : (colon). The vi editor tries to interpret any data between the first and second colon as vi editor commands. The default is nomodeline .
novice	Indicates whether you are in novice mode. You cannot change the value by using the set command.
number (nu)	Displays lines prefixed with their line numbers. The default is nonu .
optimize (opt)	Speeds the operation of terminals that lack cursor addressing. The default is noopt .
paragraphs (para=)	Defines vi macro names that start paragraphs. The default is para=IPLPPPQPP\ Lippipnppb . Single-letter nroff macros, such as the .P macro, must include the space as a quoted character if respecifying a paragraph.
partialchar (pc=)	Appears in the last display column where a double-wide character would not be displayed completely. The default character is - (minus sign).
prompt	Prompts for a new vi editor command when in command mode by printing a : (colon). The default is on .
readonly (ro)	Sets permanent read-only mode. The default is readonly .
redraw (redraw)	Simulates a smart workstation on a dumb workstation. The default is nore .
remap	Allows defining macros in terms of other macros. The default is on .
report (re=)	Sets the number of times you can repeat a command before a message is displayed. For subcommands that produce many messages, such as global subcommands, the messages are displayed when the command sequence completes. The default is report=5 .
scroll (scr=)	Sets the number of lines to be scrolled when the user scrolls up or down. The default is 1/2 of the window size, rounded down.
sections (sect=)	Defines vi macro names that start sections. The default is sect=NHSHHH\ HUuhsh+c . Single-letter nroff macros, such as the .P macro, must include the space as a quoted character if respecifying a paragraph.
shell (sh=)	Defines the shell for the ! subcommand or the !! subcommand. The default is the login shell.
shiftwidth (sw=)	Sets the distance for the software tab stops used by the autoindent option, the shift commands (> and <), and the text input commands (the Ctrl-D and Ctrl-T key sequences). This vi option only affects the indentation at the beginning of a line. The default is sw=8 .
showmatch (sm)	Shows the ((matching left parenthesis) or { (left bracket) as you type the) (right parenthesis) or } (right bracket). The default is nosm .

vi Option (Abbreviation)	Description
showmode (smd)	Displays a message to indicate when the vi editor is in input mode. The default is nosmd .
slowopen (slow)	Postpones updating the display screen during inserts. The default is noslow .
tabstop (ts=)	Sets the distance between tab stops in a displayed file. The default is ts=8 .
tags (tags =)	Defines the search path for the database file of function names created using the ctags command. The default is tags=tags\ /usr/lib/tags .
term (term=)	Sets the type of workstation you are using. The default is term=\$TERM , where \$TERM is the value of the TERM shell variable.
terse (terse)	Allows the vi editor to display the short form of messages. The default is noterse .
timeout (to)	Sets a time limit of two seconds on an entry of characters. This limit allows the characters in a macro to be entered and processed as separate characters when the timeout option is set. To resume use of the macro, set the notimeout option. The default is to .
ttytype	Indicates the tty type for the terminal being used. You cannot change this value from the vi editor.
warn (warn)	Displays a warning message before the ! subcommand executes a shell command if it is the first time you issued a shell command after changes were made in the editing buffer but not written to a file. The default is warn .
window (wi=)	Sets the number of lines displayed in one window of text. The default depends on the baud rate at which you are operating: 600 baud or less, 8 lines; 1200 baud, 16 lines; higher speeds, full screen minus 1 line.
wrapmargin (wm=)	Sets the margin for automatic word wrapping from one line to the next. The default is wm=0 . A value of 0 turns off word wrapping.
wrapscan (ws)	Allows string searches to wrap from the end of the editing buffer to the beginning. The default is ws .
wraptype (wt=)	Indicates the method used to wrap words at the end of a line. The default value is general . You can specify one of the following four values: <ul style="list-style-type: none"> general Allows wraps on word breaks as white space between two characters. This setting is the default. word Allows wraps on words. rigid Allows wraps on columns and before closing punctuation. flexible Allows wraps on columns, but one character of punctuation can extend past the margin.
writeany (wa)	Turns off the checks usually made before a write subcommand. The default is nowa .

To see a list of the vi editor settings that have changed from the default settings, enter **set** and press the spacebar. Press the Enter key to return to the command mode.

To see a complete list of the vi editor settings, enter **set all**. Press the Enter key to return to the command mode.

To turn on a vi editor option, enter **set Option**. This command automatically returns you to the command mode.

To turn on multiple vi editor options, enter `set Option Option Option`. This command turns on the three designated vi editor options and returns you to the command mode.

To turn off a vi editor option, enter `set noOption`. This command automatically returns you to the command mode.

To change the value of a vi editor option, enter `set Option=Value`. This command automatically returns you to the command mode.

You can use the `:set` subcommand of the vi editor to set options for this editing session only, or to set options for this editing session and all future editing sessions.

To set or change vi editor options *for this editing session only*, enter the `:set` subcommand from the command line.

To set vi options *for all editing sessions*, put the `:set` subcommand in the `EXINIT` environment variable in the `.profile` file (read by the shell on login) or put the `set` subcommand into a `.exrc` file. The vi editor first looks for the `EXINIT` environment variable and runs its commands. If the `EXINIT` environment variable does not exist, the vi editor then looks for the `$HOME/.exrc` file and runs its commands. Last, and regardless of any previous results, the vi editor looks for the local `.exrc` file and runs its commands.

Note: This process is true except with the `tvi` command (trusted vi). In this instance, the vi editor looks for and runs only the `/etc/.exrc` file.

For information about changing an option by setting the `EXINIT` environment variable, see the description of environment variables in the `environment` file.

The `.exrc` file can contain subcommands of the form `set Option=Value`; for example:

```
set cp=3 . . ;
```

To include a comment in the `.exrc` file, use a `"` (double quotation mark) as the first character in the line.

Defining Macros

If you use a subcommand or sequence of subcommands frequently, you can use the vi editor to define a macro that issues that subcommand or sequence.

To define a macro, enter the sequence of subcommands into a buffer named with a letter of the alphabet. The lowercase letters a through z overlay the contents of the buffer, and the uppercase letters A through Z append text to the previous contents of the buffer, allowing you to build a macro piece by piece.

For example, to define a buffer macro named c that searches for the word corner and makes the third line after the word corner the current line, enter the following command:

```
o /corner/+3
```

Then press the Esc key and enter the following command:

```
"c
```

where c is the name of the buffer macro.

To add text to the previous contents of the defined buffer, enter the `o viSubcommand`, press the Esc key, and enter `"CapitalLetter`, where the `CapitalLetter` variable specifies an uppercase letter A through Z. For example, to build a buffer macro named T that searches for the word corner and allows you to add more commands, enter the following command:

```
o corner
```

Then press the Esc key and enter the following command:

```
"T
```

where T is the name of the buffer macro. You can repeat this process at any time to add more vi subcommands to the same buffer.

For example, to add commands that move the cursor to the previous line and delete that line, enter the following command:

```
o -dd
```

where - (minus sign) means to move the cursor up one line, and dd means to delete the current line. Press the Esc key and enter the following command:

```
"Tdd
```

To start the macro, enter @Letter, where the *Letter* variable specifies the letter name of the buffer macro you want to use. To use the same macro again, enter @@ (two at symbols). For example, enter @T to start the T buffer macro and run the **search**, **move cursor**, and **delete line** commands. Enter @@T to start the T buffer macro again.

The character set used by your system is defined by the collation table. This table affects the performance of vi macros.

Mapping Keyse

You can use the **:map**, **:map!**, and **:ab** subcommands to map a keystroke to a command or a sequence of commands. The **:map** subcommand is used in the command mode. The **:map!** and **:ab** subcommands are used in the text input mode. You can map keys for this editing session and all future editing sessions or only for the current editing session from either mode.

To map keys *for all future editing sessions*, put the subcommand into a **\$HOME/.exrc** file. Each time you start the vi editor, it reads this file. The mapping remains in effect for every editing session.

To map keys *for the current editing session only* from the *command mode*, start the subcommand during the vi editor session. To map keys for the current editing session only from the *text input mode*, enter the subcommand on the command line during the vi editor session. The mapping remains in effect only for the current editing session.

Attention: If you use an IBM 3161 ASCII display station, IBM 3163 ASCII display station, or IBM 3101 ASCII display station, the default key-mapping of the vi editor can cause you to lose data. To see the default mapping, issue a **:map** subcommand. Specific problems arise with the Esc-J or Shift-J key sequence. These key sequences delete all information from the current position of the cursor to the end of the file. To avoid problems, change this key sequence using a **.exrc** file.

The **:map**, **:map!**, and **:ab** subcommands are defined and used as follows:

:map Defines macros in the command mode. The **:map** subcommand allows you to run a specified command or sequence of commands by pressing a single key while in the vi editor.

To map keys in the command mode, start the vi editor with an empty editing buffer and do not name a vi file using the **vi** command or type anything into the buffer after the vi editor starts. You can use the **:map** subcommand to do the following:

- To map a character to a sequence of editing commands, enter:
:map Letter viSubcommand
- To unmap a character previously mapped in command mode, enter:
:unmap Letter
- To display a list of current mappings for the command mode, enter
:map

The following keys are not used by the vi editor, but are available for use with the **:map** subcommand in the command mode:

- Letters g, K, q, V, and v
- Control key sequences Ctrl-A, Ctrl-K, Ctrl-O, Ctrl-W, and Ctrl-X
- Symbols _ (underscore), * (asterisk), \ (backslash), and = (equal sign)

Although you can map a key that is already used by the vi editor, the key's usual function is not available as long as the map is in effect. Some terminals allow you to map command sequences to function keys. If you are in LISP mode, the = (equal sign) cannot be used because it is used by the vi editor.

To map the letter v to the sequence of commands that would locate the next occurrence of the word map and change it to the word MAP, enter the following command:

```
:map v /map<Ctrl-V><Enter>cwMAP<Ctrl-V><Esc><Ctrl-V><Enter>
```

The previous example instructs the vi editor to locate the next occurrence of map (/map<Ctrl-V><Enter>), change map to MAP (cwMAP), end the change-word subcommand (<Ctrl-V><Esc>), and enter the command (<Ctrl-V><Enter>).

Note: To prevent the vi editor from interpreting the Enter key, it must be preceded by the Ctrl-V key sequence when being mapped. This condition is also true of the Esc, Backspace, and Delete keys.

To map the control characters Ctrl-A, Ctrl-K, and Ctrl-O, simultaneously press the Ctrl key and the letter. For example, to map the Ctrl-A key sequence to the sequence of commands that saves a file and edits the next one in a series, enter the following command:

```
:map <Ctrl-A> :w<Ctrl-V><Enter>;n<Ctrl-V><Enter>
```

To map the control characters Ctrl-T, Ctrl-W, and Ctrl-X, you must first escape them with the Ctrl-V key sequence.

To map the | (pipe symbol), you must first escape it with the two Ctrl-V key sequences, as illustrated by the following example that maps the character g to the sequence of commands that escapes to the shell, concatenates the file `/etc/motd`, and pipes the output to the `wc` command:

```
:map g :!cat /etc/motd <Ctrl-V><Ctrl-V>| wc<Ctrl-V><Enter>
```

If your terminal permits you to map function keys, you must reference them with the `#number` key sequence to designate the number of the function key that you want to map. In the following example, the F1 function key is mapped to the sequence of commands that deletes a word and moves the cursor three words down:

```
:map #1 dwww
```

In order for function key mapping to work, the output of the function key for your terminal type must match the output defined in the `terminfo` file. These definitions are denoted by the `kfnumber` entries, where `kf1` represents the F1 function key, `kf2` represents the F2 function key, and so on. If the output that you get when you press the function key does not match this entry, you must use the terminal's setup mode to correct the settings to match these terminal database entries before any mapping can occur.

You can also map certain keyboard special keys, such as the Home, End, Page Up, and Page Down keys. For most terminals, these keys are already mapped in the vi editor. You can verify this mapping by using the `:map` subcommand. If these keys are not already mapped, you can use the `:map` subcommand as follows:

```
:map <Ctrl-V><End> G
:map <Ctrl-V><Home> 1G
:map <Ctrl-V><PageUp> <Ctrl-F>
:map <Ctrl-V><PageDown> <Ctrl-B>
```

To get a listing of all current maps in the command mode, enter the `:map` subcommand. The preceding examples are then displayed as follows:

```
v          v          /map<Ctrl-M>cwMAP<Ctrl-[]><Ctrl-M>
<Ctrl-A> <Ctrl-A>    :w<Ctrl-M>:n<Ctrl-M>
g          g          :!cat /etc/motd | wc <Ctrl-M>
```

Note: The Ctrl-V and Enter key sequence is displayed as the Ctrl-M key sequence, and the Ctrl-V and Esc key sequence is displayed as the Ctrl-[key sequence.

:map! Maps character strings to single keys while in text input mode. To map keys in the text input mode, start the vi editor with an empty editing buffer and do not name a vi file using the `vi` command or type anything into the buffer after the vi editor starts. You can use the `:map!` subcommand to do the following:

- To map a letter to one or more vi strings in text input mode, enter:
:map! Letter String
- To unmap a letter previously mapped in text input mode, enter:
:unmap! Letter
- To display a list of existing strings that are mapped to specific keys in text input mode, enter:
:map!

Typing the mapped key while in text input mode produces the specified string. The Ctrl-V and Esc key sequence puts you into command mode, backs up to the beginning of the current word (**bbw**), and starts the `cw` (change-word) subcommand. For example:

```
:map! % <Ctrl-V><Esc>bbwcw
```

When typing text, if you realize that you have mistyped a word, you can change it by pressing the % (percent) key and retyping the word. You are automatically returned to insert mode.

Note: Be careful when choosing keys to be used for the `:map!` subcommand. Once keys have been mapped, they can no longer be input as text without first issuing the `:unmap!` subcommand.

:ab Maps a key or sequence of keys to a string of characters for use in the text input mode. The **:ab** subcommand is useful when inputting text that possesses several repetitive phrases, names, or titles.

The following example replaces the word `city` with the phrase `Austin, Texas 78759` whenever it is typed in text input mode and followed by a white space, period, or comma:

```
:ab city Austin, Texas 78759
```

For example, if while inputting text, you type the following:

```
My current residence is city.
```

Pressing the Tab key expands the word `city` to read:

```
My current residence is Austin, Texas 78759.
```

The abbreviation is not expanded within a word. For example, if you type `My current residence iscity`, the word `iscity` is not expanded.

If the **:map!** subcommand is used to map abbreviations for insert mode, then all occurrences of the abbreviations are expanded regardless of where it occurs. If you used the **:map!** subcommand for the preceding example (**:map!** `city Austin, Texas 78759`), then whenever you type the word `city`, regardless of what precedes or follows, the word will be expanded to `Austin, Texas 78759`. Therefore, the word `iscity` becomes `isAustin, Texas 78759`.

Note: Be careful when choosing the keys that are used for the **:ab** subcommand. Once keys are defined, they can no longer be input as text without first issuing the **:unab** subcommand.

Setting Abbreviations

The **set** command has behavior similar to the **map!** command except that the **set** command substitutes the string for the abbreviation only when the abbreviation is a separate word. You can use the **set** command of the vi editor to:

- List existing abbreviations
- Remove an abbreviation
- Set (define) an abbreviation

Note: Start the vi editor with an empty editing buffer. Do not name a vi file using the **vi** command or type anything into the buffer after the vi editor starts. Press the Esc key to be sure you are in the command mode.

To list abbreviations

Enter the **:ab** command to list existing abbreviations. Press the Enter key to return to command mode.

To remove abbreviations

Enter the **:anab***Abbreviation* command to remove an abbreviation, where the *Abbreviation* variable specifies the character string you do not want abbreviated any more.

To set (define) an abbreviation

Enter the **:ab** *Abbreviation String* command to set an abbreviation, where the *Abbreviation* variable specifies the character string being defined as an abbreviation and the *String* variable specifies the character string being abbreviated. The abbreviation can be substituted for the string only when the abbreviation is a separate word.

For example, if you enter the **:ab kn upper** command and then type `acknowledge` while in the text input mode, the set abbreviation string is not started because the `kn` string in the word `acknowledge` is not a separate word.

However, if you type the **:ab kn upper** command and then type `make the kn line all kncase` while in the text input mode, the result is `make the upper line all uppercase`.

Flags

-c <i>Subcommand</i>	Carries out the ex editor subcommand before viewing with vi begins. The cursor moves to the line affected by the last subcommand to be carried out. When a null operand is entered, as in -c' , the vi editor places the cursor on the first line of the file. The -c flag is incompatible with the + flag. Do not specify both flags at the same time.
-l	Enters the vi editor in LISP mode. In this mode, the vi editor creates indents appropriate for LISP code, and the (,), {, }, [[, and]] subcommands are modified to act appropriately for LISP.
-r [<i>File</i>]	Recovers a file after a vi editor or system malfunction. If you do not specify the <i>File</i> variable, the vi editor displays a list of all saved files.
-R	Sets the readonly option to protect the file against overwriting.
-t <i>Tag</i>	Edits the file containing the <i>Tag</i> variable and positions the vi editor at its definition. To use this flag, you must first create a database of function names and their locations using the ctags command.
-v	Enters the vi editor in the verbose mode.
-w <i>Number</i>	Sets the default window size to the value specified by the <i>Number</i> variable. This flag is useful when you use the vi editor over a low-speed line.
-y <i>Number</i>	Overrides the maximum line setting of 1,048,560 with any value greater than 1024. You should request twice the number of lines that you require because the vi editor uses the extra lines for buffer manipulation.
+ [<i>Subcommand</i>]	Carries out the ex editor subcommand before editing begins. If you do not specify the <i>Subcommand</i> variable, the cursor is placed on the first line of the file. This + flag is incompatible with the -c flag. Do not specify both flags at the same time.

vi General Subcommand Syntax

Use the following general syntax to enter subcommands:

[*Named_Buffer*] [*Operator*] [*Number*] *Object*

Note: Square brackets indicate optional items.

[<i>Named_Buffer</i>]	Specifies a temporary text storage area.
[<i>Operator</i>]	Specifies the subcommand or action; instructs the vi editor.
[<i>Number</i>]	Specifies either the extent of the action or a line address as a whole number.

Object Specifies what to act on, such as a text object (a character, word, sentence, paragraph, section, character string) or a text position (a line, position in the current line, screen position).

Counts before Subcommands

You can put a number in front of many subcommands. The vi editor interprets this number in one of the following ways:

- Go to the line specified by the *Number* parameter:
5G
10Z
- Go to the column specified by the *Number* parameter:
25|
- Scroll the number of lines up or down specified by the *Number* parameter:
10Ctrl-U
10Ctrl-D

vi Editor Subcommands

Use the subcommands to perform these kinds of actions:

- Moving the cursor
- Editing text
- Manipulating files
- Other actions

Moving the Cursor

Use subcommands to move the cursor within a file in these ways:

- Moving within a line
- Moving within a line by character position
- Moving to words
- Moving by line position
- Moving to sentences, paragraphs, or sections
- Moving by redrawing the screen
- Paging and scrolling
- Searching for patterns
- Marking a specific location in a file and returning

Moving within a Line

Enter the following subcommands in command mode. You can cancel an incomplete command by pressing the Esc key. If you need information about the format of vi subcommands, see "vi General Subcommand Syntax."

Left Arrow or **h** or **Ctrl-H**

Down Arrow or **j** or **Ctrl-J** or **Ctrl-N**

Up Arrow or **k** or **Ctrl-P**

Right Arrow or **l**

Moves the cursor one character to the left.

Moves the cursor down one line (it remains in the same column).

Moves the cursor up one line (it remains in the same column).

Moves the cursor one character to the right.

Moving within a Line by Character Position

Enter the following subcommands in command mode. You can cancel an incomplete command by pressing the Esc key. If you need information about the format of vi subcommands, see "vi General Subcommand Syntax."

^	Moves the cursor to the first nonblank character.
0	Moves the cursor to the beginning of the line.
\$	Moves the cursor to the end of the line.
fx	Moves the cursor to the next <i>x</i> character.
Fx	Moves the cursor to the last <i>x</i> character.
tx	Moves the cursor to one column before the next <i>x</i> character.
Tx	Moves the cursor to one column after the last <i>x</i> character.
;	Repeats the last f , F , t , or T subcommand.
,	Repeats the last f , F , t , or T subcommand in the opposite direction.
<i>Number</i> 	Moves the cursor to the specified column.

Moving to Words

Enter the following subcommands in command mode. If you need information about the format of vi subcommands, "vi General Subcommand Syntax."

w	Moves the cursor to the next small word.
b	Moves the cursor to the previous small word.
e	Moves the cursor to the next end of a small word.
W	Moves the cursor to the next big word.
B	Moves the cursor to the previous big word.
E	Moves the cursor to the next end of a big word.

Moving by Line Position

Enter the following subcommands in command mode. If you need information about the format of vi subcommands, see "vi General Subcommand Syntax."

H	Moves the cursor to the top line on the screen.
L	Moves the cursor to the last line on the screen.
M	Moves the cursor to the middle line on the screen.
+	Moves the cursor to the next line at its first nonblank character.
-	Moves the cursor to the previous line at its first nonblank character.
Enter	Moves the cursor to the next line at its first nonblank character.

Moving to Sentences, Paragraphs, or Sections

Enter the following subcommands in command mode. You can cancel an incomplete subcommand by pressing the Esc key. If you need information about the format of vi subcommands, see "vi General Subcommand Syntax."

(Places the cursor at the beginning of the previous sentence, or the previous s-expression if you are in LISP mode.
)	Places the cursor at the beginning of the next sentence, or the next s-expression if you are in LISP mode.
{	Places the cursor at the beginning of the previous paragraph, or at the next list if you are in LISP mode.
}	Places the cursor at the beginning of the next paragraph, at the next section if you are in C mode, or at the next list if you are in LISP mode.
]]	Places the cursor at the next section, or function if you are in LISP mode.

[] Places the cursor at the previous section, or function if you are in LISP mode.

Moving by Redrawing the Screen

Enter the following subcommands in command mode. You can cancel an incomplete subcommand by pressing the Esc key. If you need information about the format of vi subcommands, see "vi General Subcommand Syntax."

z	Redraws the screen with the current line at the top of the screen.
z-	Redraws the screen with the current line at the bottom of the screen.
z.	Redraws the screen with the current line at the center of the screen.
/Pattern/z-	Redraws the screen with the line containing the character string, specified by the <i>Pattern</i> parameter, at the bottom.

Paging and Scrolling

Enter the following subcommands in command mode. You can cancel an incomplete subcommand by pressing the Esc key. If you need information about the format of vi subcommands, see "vi General Subcommand Syntax."

Ctrl-U	Scrolls up one-half screen.
Ctrl-D	Scrolls down one-half screen.
Ctrl-F	Scrolls forward one screen.
Ctrl-B	Scrolls backward one screen.
Ctrl-E	Scrolls the window down one line.
Ctrl-Y	Scrolls the window up one line.
z+	Pages up.
z^	Pages down.

Searching for Patterns

Enter the following subcommands in command mode. You can cancel an incomplete subcommand by pressing the Esc key. If you need information about the format of vi subcommands, see "vi General Subcommand Syntax."

[Number]G	Places the cursor at the line number specified by the <i>Number</i> parameter or at the last line if the <i>Number</i> parameter is not specified.
/Pattern	Places the cursor at the next line containing the character string specified by the <i>Pattern</i> parameter.
?Pattern	Places the cursor at the next previous line containing the character string specified by the <i>Pattern</i> parameter.
n	Repeats the last search for the text specified by the <i>Pattern</i> parameter in the same direction.
N	Repeats the last search for the text specified by the <i>Pattern</i> parameter in the opposite direction.
/Pattern/+Number	Places the cursor the specified number of lines after the line matching the character string specified by the <i>Pattern</i> parameter.
?Pattern?-Number	Places the cursor the specified number of lines before the line matching the character string specified by the <i>Pattern</i> parameter.
%	Finds the parenthesis or brace that matches the one at current cursor position.

Editing Text

The subcommands for editing enable you to perform the following tasks:

- Marking a specific location in a file and returning
- Adding text to a file
- Changing text while in input mode
- Changing text from command mode
- Copying and moving text
- Restoring and repeating changes

Marking a Specific Location in a File and Returning

Enter the following subcommands in command mode. You can cancel an incomplete subcommand by pressing the Esc key. If you need information about the format of vi subcommands, see "vi General Subcommand Syntax."

- " Moves the cursor to the previous location of the current line.
- " Moves the cursor to the beginning of the line containing the previous location of the current line.
- mx** Marks the current position with the letter specified by the *x* parameter.
- ``x` Moves the cursor to the mark specified by the *x* parameter.
- `'x` Moves the cursor to the beginning of the line containing the mark specified by the *x* parameter.

Adding Text to a File (Text Input Mode)

Enter the following subcommands in command mode to change the vi editor into text input mode. If you need information about the format of vi subcommands, see "vi General Subcommand Syntax."

- aText** Inserts text specified by the *Text* parameter after the cursor. End text input mode by pressing the Esc key.
- AText** Adds text specified by the *Text* parameter to the end of the line. End text input mode by pressing the Esc key.
- iText** Inserts text specified by the *Text* parameter before the cursor. End text input mode by pressing the Esc key.
- IText** Inserts text specified by the *Text* parameter before the first nonblank character in the line. End text input mode by pressing the Esc key.
- o** Adds an empty line below the current line. End text input mode by pressing the Esc key.
- O** Adds an empty line above the current line. End text input mode by pressing the Esc key.

Changing Text While in Input Mode

Use the following subcommands only while in text input mode. These commands have different meanings in command mode. If you need information about the format of vi subcommands, see "vi General Subcommand Syntax."

- Ctrl-D** Goes back to previous autoindent stop.
- ^ Ctrl-D** Ends autoindent for this line only.
- 0Ctrl-D** Moves cursor back to left margin.
- Esc** Ends insertion and returns to command state.
- Ctrl-H** Erases the last character.
- Ctrl-Q** Enters any character if xon is disabled.
- Ctrl-V** Enters any character.
- Ctrl-W** Erases the last small word.

<code>\</code>	Quotes the erase and kill characters.
<code>Ctrl-?</code>	Interrupts and ends insert or the Ctrl-D key sequence.

Changing Text from Command Mode

Use the following subcommands in command mode. An incomplete subcommand can be canceled by pressing the Esc key. If you need information about the format of vi subcommands, see "vi General Subcommand Syntax."

<code>C</code>	Changes the rest of the line (same as <code>c\$</code>).
<code>C</code>	Changes a line.
<code>cc</code>	Changes a word.
<code>cw</code>	Changes a word.
<code>cwText</code>	Changes a word to the text specified by the <i>Text</i> parameter.
<code>D</code>	Deletes the rest of the line (same as <code>d\$</code>).
<code>dd</code>	Deletes a line.
<code>dw</code>	Deletes a word.
<code>J</code>	Joins lines.
<code>rx</code>	Replaces the current character with the character specified by <i>x</i> .
<code>RText</code>	Overwrites characters with the text specified by the <i>Text</i> parameter.
<code>s</code>	Substitutes characters (same as <code>cl</code>).
<code>S</code>	Substitutes lines (same as <code>cc</code>).
<code>u</code>	Undoes the previous change.
<code>x</code>	Deletes a character at the cursor.
<code>X</code>	Deletes a character before the cursor (same as <code>dh</code>).
<code><<</code>	Shifts one line to the left.
<code><L</code>	Shifts all lines from the cursor to the end of the screen to the left.
<code>>></code>	Shifts one line to the right.
<code>>L</code>	Shifts all lines from the cursor to the end of the screen to the right.
<code>~</code>	Changes letter at the cursor to the opposite case.
<code>!</code>	Indents for LISP.

Copying and Moving Text

Use the following subcommands in command mode. An incomplete subcommand can be canceled by pressing the Esc key. If you need information about the format of vi subcommands, see "vi General Subcommand Syntax."

<code>p</code>	Puts back text from the undo buffer after the cursor.
<code>P</code>	Puts back text from the undo buffer before the cursor.
<code>"xp</code>	Puts back text from the <i>x</i> buffer.
<code>"xd</code>	Deletes text into the <i>x</i> buffer.
<code>y</code>	Places the object that follows (for example, <code>w</code> for word) into the undo buffer.
<code>"xy</code>	Places the object that follows into the <i>x</i> buffer, where <i>x</i> is any letter.
<code>Y</code>	Places the line in the undo buffer.

Restoring and Repeating Changes

Use the following subcommands in command mode. An incomplete subcommand can be canceled by pressing the Esc key. If you need information about the format of vi subcommands, see "vi General Subcommand Syntax."

u Undoes the last change.

Note: After an undo, the cursor moves to the first non-blank character on the updated current line.

U Restores the current line if the cursor has not left the line since the last change.

. Repeats the last change or increments the "*np* command.

Note:

1. This subcommand will repeat the last change, including an undo. Therefore, after an undo, repeat performs an undo rather than repeat the last change.

2. This subcommand is not meant for use with a macro. Enter @@ (two at signs) to repeat a macro.

"n p Retrieves the *n*th last delete of a complete line or block of lines.

Manipulating Files

The subcommands for manipulating files allow you to do the tasks outlined in the following sections:

- Saving changes to a file
- Editing a second file
- Editing a list of files
- Finding file information

Saving Changes to a File

Use the following subcommands in command mode. If you need information about the format of vi subcommands, see "vi General Subcommand Syntax."

:w Writes the edit buffer contents to the original file. If you are using this subcommand within the ex editor, you do not need to type the : (colon).

:w File Writes the edit buffer contents to the file specified by the *File* parameter. If you are using this subcommand within the ex editor, you do not need to type the : (colon).

:w! File Overwrites the file specified by the *File* parameter with the edit buffer contents. If you are using this subcommand within the ex editor, you do not need to type the : (colon).

Editing a Second File

Enter the following subcommands in command mode. If you need information about the format of vi subcommands, see "vi General Subcommand Syntax."

:e File Edits the specified file. If you are using this subcommand from the ex editor, you do not need to type the : (colon).

:e! Re-edits the current file and discards all changes.

:e + File Edits the specified file starting at the end.

:e + Number File Edits the specified file starting at the specified line number.

:e # Edits the alternate file. The alternate file is usually the previous file name before accessing another file with a **:e** command. However, if changes are pending on the current file when a new file is called, the new file becomes the alternate file. This subcommand is the same as the **Ctrl-A** subcommand.

:r File Reads the file into the editing buffer by adding new lines below the current line. If you are using this subcommand from the ex editor, you do not need to type the : (colon).

:r !Command Runs the specified command and places its output into the file by adding new lines below the current cursor position.

:ta Tag Edits a file containing the *Tag* tag starting at the location of the tag. To use this subcommand, you must first create a database of function names and their locations using the **ctags** command. If you are using this subcommand from the ex editor, you do not need to type the : (colon).

Ctrl-]	Edits a file containing the tag associated with the current word starting at the location of the tag. To use this subcommand, you must first create a database of function names and their locations using the ctags command. Ctrl-T edits a file at the editing position where the previous Ctrl-] subcommand was issued. If multiple Ctrl-] subcommands have been issued, then multiple Ctrl-T subcommands can be used to return to previous editing positions where Ctrl-] subcommands were issued.
Ctrl-A	Edits the alternate file. The alternate file is usually the previous current file name. However, if changes are pending on the current file when a new file is called, the new file becomes the alternate file. This subcommand is the same as the :e # subcommand.

Editing a List of Files

Enter the following subcommands in command mode. If you need information about the format of vi subcommands, see "vi General Subcommand Syntax."

:n	Edits the next file in the list entered on the command line. If you are using this subcommand from the ex editor, a : (colon) is not needed.
:n Files	Specifies a new list of files to edit. If you are using this subcommand from the ex editor, a : (colon) is not needed.

Finding File Information

Enter the following subcommand in command mode. If you need information about the format of vi subcommands, see "vi General Subcommand Syntax".

Ctrl-G	Shows the current file name, current line number, number of lines in the file, and percentage of the way through the file where the cursor is located.
---------------	--

Other Actions

The vi editor provides the subcommands described in the following sections:

- Adjusting the screen
- Entering shell commands
- Interrupting and ending the vi editor

Adjusting the Screen

Enter the following subcommands in command mode. An incomplete subcommand can be canceled by pressing the Esc key. If you need information about the format of vi subcommands, see "vi General Subcommand Syntax."

Ctrl-L	Clears and redraws the screen.
Ctrl-R	Redraws the screen and eliminates blank lines marked with @ (at sign).
zNumber	Makes the window the specified number of lines long.

Entering Shell Commands

The following subcommands allow you to run a command within the vi editor. Enter these subcommands in command mode. If you need information about the format of vi subcommands, see "vi General Subcommand Syntax."

:sh	Enters the shell to allow you to run more than one command. You can return to the vi editor by pressing the Ctrl-D key sequence. If you are using this subcommand within the ex editor, a : (colon) is not needed.
!:Command	Runs the specified command and then returns to the vi editor. If you are using this subcommand within the ex editor, a : (colon) is not needed.
	Note: The # (alternate file), % (current file), and ! (previous command) special characters are expanded when following a !: subcommand. To prevent any of these characters from being expanded, use the \ (backslash).
:!!	Repeats the last !:Command subcommand.
Number!!Command	Runs the specified command and replaces the lines specified by <i>Number</i> with the output of the command. If a number is not specified, the default value is 1. If the command expects standard input, the specified lines are used as input.
!Object Command	Runs the specified command and replaces the object specified by the <i>Object</i> parameter with the output of the command. If the command expects standard input, the specified object is used as input.

Interrupting and Ending the vi Editor

Enter the following subcommands in command mode. If you need information about the format of vi subcommands, see "vi General Subcommand Syntax."

Q	Enters the ex editor in command mode.
ZZ	Exits the vi editor, saving changes.
:q	Quits the vi editor. If you have changed the contents of the editing buffer, the vi editor displays a warning message and does not quit. If you are using this subcommand from the ex editor, a : (colon) is not needed.
:q!	Quits the vi editor, discarding the editing buffer. If you are using this subcommand from the ex editor, a : (colon) is not needed.
Esc	Ends text input or ends an incomplete subcommand.
Ctrl-?	Interrupts a subcommand.

Exit Status

The following exit values are returned:

0	Indicates successful completion.
>0	Indicates an error occurred.

Input Files

Input files must be text files or files that are similar to text files except for an incomplete last line that is no longer than 8191 bytes in length and contains no null characters.

The **.exrc** files must be text files consisting of **ex** commands.

By default, the vi editor reads lines from the files to be edited without interpreting any of those lines as any form of vi editor command.

Related Information

sed command.

viosecure command

Purpose

Activates, deactivates, and displays security hardening rules. Configures, unconfigures or displays firewall settings.

Syntax

```
viosecure -level LEVEL [-apply] | [-nonint] -view
```

```
viosecure -firewall on [[-force] -reload]
```

```
viosecure -firewall allow | deny -port number [-interface ifname] [-address IPaddress] [-timeout Timeout]  
[-remote]
```

```
viosecure -firewall view [-fmt delimiter]
```

```
viosecure -undo
```

Description

The **viosecure** activates, deactivates, and displays security hardening rules. By default, none of the security hardening features are activated after installation. Upon running the **viosecure** command, the command guides the user through the proper security settings, which range from High to Medium to Low. After this initial selection, a menu is displayed itemizing the security configuration options associated with the selected security level in sets of 10. These options can be accepted in whole, individually toggled off or on, or ignored. After any changes, **viosecure** continues to apply the security settings to the computer system.

The **viosecure** command also configures, unconfigures, and displays network firewall settings. Using the **viosecure** command, you can activate and deactivate specific ports and specify the interface and IP address from which connections will be allowed.

Flags

-apply	Applies all of the <i>LEVEL</i> security settings to the system. There is no user-selectable option.
-firewall allow -port <i>Port</i> [-interface <i>ifname</i>] [-address <i>IPaddress</i>] [-timeout <i>Timeout</i>] [-source]	Permits IP activity per port with optional parameters according to interface, IP address, and time that it is effective. The <i>Port</i> argument can be a number or a service name from the <i>/etc/services</i> file. The <i>remote</i> option specifies that the port is a remote port. All IP activity to and from that remote port is allowed. The default is all IP activity to and from a local port is allowed. The timeout period can be specified as a number (in seconds), or with a number followed by <i>m</i> (minutes), <i>h</i> (hours), or <i>d</i> (days). The maximum timeout period is 30 days.
-firewall deny -port <i>Port</i> [-interface <i>ifname</i>] [-address <i>IPaddress</i>] [-timeout <i>Timeout</i>] [-source]	Removes a previous firewall -allow setting. The <i>Port</i> argument can be a number or a service name from the <i>/etc/services</i> file. If -port 0 is specified, then all allow settings are removed. The <i>remote</i> option specifies that the port is the remote port. The default is local port. The timeout period can be specified as a number (in seconds), or with a number followed by <i>m</i> (minutes), <i>h</i> (hours) or <i>d</i> (days). The maximum timeout period is 30 days.
-firewall off	Unconfigures the firewall settings and saves all the firewall filter rules to the <i>/home/padmin/viosfirewall.rules</i> file.
-firewall on [[-force] -reload]	Configures the default firewall settings from the filter rules in ODM. If you use the reload option, then the ODM rules are deleted and the default values are loaded from the <i>/home/ios/security/viosecur.ctl</i> file. If the <i>viousecure.ctl</i> file does not exist, the force option specifies to use the hard-coded, default firewall settings.

-level LEVEL	Specifies the security LEVEL settings to choose, where LEVEL is low, medium, high, or default. The default LEVEL deactivates any previous security LEVEL system settings. Except for the default LEVEL, ten security LEVEL settings are displayed at a time. The user then can choose the desired security settings by entering comma separated numbers, the word ALL to choose all of the settings, the word NONE to choose none of the settings, the letter q to exit, or the letter h for help. The security settings chosen are then applied to the system.
-firewall view [-fmt delimiter]	Displays the current allowable ports. If the -fmt option is specified, then it divides output by a user-specified delimiter.
-nonint	Specifies non-interactive mode.
-view	Displays the current security level settings. All of the security setting names end with three characters X1s where X = l(low), m(medium), h(high) or d(default). For example, the security level name min1en11s is the low level security setting for minimum length of a password.
-undo	Removes the security settings that have been applied.

Examples

- To display the high system security settings, and to select which of the high security settings to apply to the system, type:
viosecure -level high
- To apply all of the 'high' system security settings to the system, type:
viosecure -level high -apply
- To display the current system security settings, type:
viosecure -view
- To unconfigure the previous system security settings, type:
viosecure -level default
- To allow IP activity on the ftp-data, ftp, ssh, www, https, rmc, and cimon ports, and to deny other IP activity, type:
viosecure -firewall on
- To allow IP activity on all ports, type:
viosecure -firewall off
- To allow users from IP address 10.10.10.10 to rlogin, type:
viosecure -firewall allow -port login -address 10.10.10.10
- To allow users to rlogin for seven days, type:
viosecure -firewall allow -port login -timeout 7d
- To allow rsh client activity through interface en0, type:
viosecure -firewall allow -port 514 -interface en0 -remote
- To removes the rule that allows users from IP address 10.10.10.10 to rlogin, type:
viosecure -firewall deny -port login -address 10.10.10.10
- To display the list of allowed ports, type:
viosecure -firewall view
- To undo the security settings that have been applied, type:
viosecure -undo /etc/security/aixpert/core/undo.xml

Note: This command removes all the security settings specified in the undo.xml file.

viostat command

Purpose

Reports Central Processing Unit (CPU) statistics, asynchronous input/output (AIO) and input/output statistics for the entire system, adapters, tty devices, disks and CD-ROMs.

Syntax

viostat

viostat [**-sys**] [**-adapter**] [**-tty** | **-disk**] [**-path**] [**-time**] [*PhysicalVolume ...*] [*Interval* [*Count*]]

Description

The **viostat** command is used for monitoring system input/output device loading by observing the time the physical disks are active in relation to their average transfer rates. The **viostat** command generates reports that can be used to change system configuration to better balance the input/output load between physical disks and adapters.

The first report generated by the **viostat** command provides statistics concerning the time since the system was booted. Each subsequent report covers the time since the previous report. All statistics are reported each time the **viostat** command is run. The report consists of a tty and CPU header row followed by a row of tty and CPU statistics. On multiprocessor systems, CPU statistics are calculated system-wide as averages among all processors.

If the **-sys** flag is specified, a system-header row is displayed followed by a line of statistics for the entire system. The hostname of the system is printed in the system-header row.

If the **-adapter** flag is specified, an adapter-header row is displayed followed by a line of statistics for the adapter. This will be followed by a disk-header row and the statistics of all the disks/CD-ROMs connected to the adapter. Such reports are generated for all the disk adapters connected to the system.

A disks header row is displayed followed by a line of statistics for each disk that is configured. If the *PhysicalVolume* parameter is specified, only those names specified are displayed.

If the *PhysicalVolume* parameter is specified, one or more alphabetic or alphanumeric physical volumes can be specified. If the *PhysicalVolume* parameter is specified, the tty and CPU reports are displayed and the disk report contains statistics for the specified drives. If a specified drive name is not found, the report lists the specified name and displays the message Drive Not Found. If no Drive Names are specified, the report contains statistics for all configured disks and CD-ROMs. If no drives are configured on the system, no disk report is generated. The first character in the *PhysicalVolume* parameter cannot be numeric.

The *Interval* parameter specifies the amount of time in seconds between each report. The first report contains statistics for the time since system startup (boot). Each subsequent report contains statistics collected during the interval since the previous report. The *Count* parameter can be specified in conjunction with the *Interval* parameter. If the *Count* parameter is specified, the value of count determines the number of reports generated at *Interval* seconds apart. If the *Interval* parameter is specified without the *Count* parameter, the **viostat** command generates reports continuously.

The **viostat** command reports number of physical processors consumed (*physc*) and the percentage of entitlement consumed (*% entc*) in shared processor environments. These metrics will only be displayed in shared processor environments.

The **viostat** command is useful in determining whether a physical volume is becoming a performance bottleneck and if there is potential to improve the situation. The % utilization field for the physical volumes indicates how evenly the file activity is spread across the drives. A high % utilization on a physical volume is a good indication that there may be contention for this resource. Since the CPU utilization statistics are also available with the viostat report, the percentage of time the CPU is in I/O wait can be determined at the same time. Consider distributing data across drives if the I/O wait time is significant and the disk utilization is not evenly distributed across volumes.

Reports

The viostat command generates four types of reports, the tty and CPU Utilization report, the Disk Utilization report, the System throughput report and the Adapter throughput report.

tty and CPU Utilization Report

The first report generated by the viostat command is the tty and CPU Utilization Report. For multiprocessor systems, the CPU values are global averages among all processors. Also, the I/O wait state is defined system-wide and not per processor. The report has the following format:

Statistic	Description
tin	Shows the total number of characters read by the system for all ttys.
tout	Shows the total number of characters written by the system to all ttys.
% user	Shows the percentage of CPU utilization that occurred while executing at the user level (application).
% sys	Shows the percentage of CPU utilization that occurred while executing at the system level (kernel).
% idle	Shows the percentage of time that the CPU or CPUs were idle and the system did not have an outstanding disk I/O request.
% iowait	Shows the percentage of time that the CPU or CPUs were idle during which the system had an outstanding disk I/O request.

This information is updated at regular intervals by the kernel (typically sixty times per second). The tty report provides a collective account of characters per second received from all terminals on the system as well as the collective count of characters output per second to all terminals on the system.

Disk Utilization Report

The second report generated by the viostat command is the Disk Utilization Report. The disk report provides statistics on a per physical disk basis. The report has a format similar to the following:

Statistic	Description
% tm_act	Indicates the percentage of time the physical disk was active (bandwidth utilization for the drive).
Kbps	Indicates the amount of data transferred (read or written) to the drive in KB per second.
tps	Indicates the number of transfers per second that were issued to the physical disk. A transfer is an I/O request to the physical disk. Multiple logical requests can be combined into a single I/O request to the disk. A transfer is of indeterminate size.
Kb_read	The total number of KB read.
Kb_wrtn	The total number of KB written.

Statistics for CD-ROM devices are also reported.

For large system configurations where a large number of disks are configured, the system can be configured to avoid collecting physical disk input/output statistics when the **viostat** command is not executing. If the system is configured in the above manner, the first Disk report displays the message `Disk History Since Boot Not Available` instead of the disk statistics. Subsequent interval reports generated by the **viostat** command contain disk statistics collected during the report interval. Any tty and CPU statistics after boot are unaffected.

System Throughput Report

This report is generated if the `-sys` flag is specified. This report provides statistics for the entire system. This report has the following format:

Statistic	Description
Kbps	Indicates the amount of data transferred (read or written) in the entire system in KB per second.
tps	Indicates the number of transfers per second issued to the entire system.
Kb_read	The total number of KB read from the entire system.
Kb_wrtn	The total number of KB written to the entire system.

Adapter Throughput Report

This report is generated if the `-adapter` flag is specified. This report provides statistics on an adapter-by-adapter basis. This report has the following format:

Statistic	Description
Kbps	Indicates the amount of data transferred (read or written) in the adapter in KB per second.
tps	Indicates the number of transfers per second issued to the adapter.
Kb_read	The total number of KB read from the adapter.
Kb_wrtn	The total number of KB written to the adapter.

Disk Input/Output History

To improve performance, the collection of disk input/output statistics has been disabled. To enable the collection of this data, type:

```
chdev -dev sys0 -attr iostat=true
```

To display the current settings, type:

```
lsdev -dev sys0 -attr iostat
```

If the collection of disk input/output history is disabled, the first disk report of **viostat** output displays the message `Disk History Since Boot Not Available` instead of disk statistics. As before, subsequent interval reports generated by the **viostat** command contain disk statistics collected during the report interval.

Flags

-adapter	Displays the adapter throughput report. If the -adapter flag is specified with the -tty flag, the tty and CPU report is displayed, followed by the adapter throughput report. Disk Utilization reports of the disks connected to the adapters, will not be displayed after the Adapter throughput report. If the -adapter flag is specified with the -disk flag, tty and CPU report will not be displayed. If Physical Volume parameter is specified, the Disk Utilization Report of the specified Physical volume will be printed under the corresponding adapter to which it belongs.
-disk -path	The -disk flag is exclusive of the -tty flag and displays only the disk utilization report. The -m flag will print the path statistics for the following <ul style="list-style-type: none">• Paths to MPIO (Multi-Path I/O) enabled devices.• Paths in the ESS machines. The throughput is per device. The throughout for all the paths to that device follow the throughput of that device. For ESS machines, the vpaths will be treated as disks and hdisks will be treated as Paths. Internally the vpaths are actually disks and hdisks are the paths to them. For MPIO enabled devices, the path name will be represented as Path0, Path1, Path2 and so on. The numbers 0, 1, 2, and so on are the path IDs provided by the lspath command. Since paths to a device can be attached to any adapter, the adapter report will report the path statistics under each adapter. The disk name will be a prefix to all the paths. For all MPIO enabled devices, the adapter report will print the path names as hdisk10_Path0, hdisk0_Path1, and so on. For all ESS Machines, the adapter report will print the path names as vpath0_hdisk3, vpath10_hdisk25, and so on.
-sys	Displays the system throughput report.
-time	Prints the time-stamp next to each line of output of viostat. The time-stamp displays in the HH:MM:SS format.
-tty	The -tty flag is exclusive of the -disk flag and displays only the tty and cpu usage reports.

Examples

1. To display a single history since boot report for all tty, CPU, and Disks, type:
viostat
2. To display a continuous disk report at two second intervals for the disk with the logical name disk1, type:
viostat -disk disk1 2
3. To display six reports at two second intervals for the disk with the logical name disk1, type:
viostat disk1 2 6
4. To display six reports at two second intervals for all disks, type:
viostat -disk 2 6
5. To display six reports at two second intervals for three disks named disk1, disk2, disk3, type:
viostat disk1 disk2 disk3 2 6
6. To print the System throughput report, type:
viostat -sys
7. To print the Adapter throughput report, type:
viostat -adapter
8. To print the System and Adapter throughput reports, with only the tty and CPU report (no disk reports), type:
viostat -sys -adapter -tty

9. To print the System and Adapter throughput reports with the Disk Utilization reports of hdisk0 and hdisk7, type


```
viostat -sys -adapter -disk hdisk0 hdisk7
```
10. To display time stamp next to each line of output of viostat, type:


```
viostat -time
```

vmstat command

Purpose

Reports statistics about kernel threads, virtual memory, disks, traps, and processor activity.

Syntax

To report virtual memory statistics:

```
vmstat [ -f ] [-i ][-s ][-I ][-t ][-v ][-w ][-l ][{-p | -P | } pagesize | ALL ] ALL] [PhysicalVolume...] [Interval [Count ] ]
```

Description

The **vmstat** command reports statistics about kernel threads, virtual memory, disks, traps and CPU activity. Reports generated by the **vmstat** command can be used to balance system load activity. These system-wide statistics (among all processors) are calculated as averages for values expressed as percentages, and as sums otherwise.

Flags

- f** Reports the number of forks since system startup.
- i** Displays the number of interrupts taken by each device since system startup.
Note: The **-I**, **-t**, **-w**, and **-l** flags are ignored when they are specified with the **-i** flag.
- s** Writes to standard output the contents of the sum structure, which contains an absolute count of paging events since system initialization. The **-s** flag can only be used with the **-v** flag. These events are described as follows:
 - address translation faults**
Incremented for each occurrence of an address translation page fault. I/O may or may not be required to resolve the page fault. Storage protection page faults (lock misses) are not included in this count.
 - page ins**
Incremented for each page read in by the virtual memory manager. The count is incremented for page ins from page space and file space. Along with the page out statistic, this represents the total amount of real I/O initiated by the virtual memory manager.
 - page outs**
Incremented for each page written out by the virtual memory manager. The count is incremented for page outs to page space and for page outs to file space. Along with the page in statistic, this represents the total amount of real I/O initiated by the virtual memory manager.
 - paging space page ins**
Incremented for VMM initiated page ins from paging space only.

total reclaims

Incremented when an address translation fault can be satisfied without initiating a new I/O request. This can occur if the page has been previously requested by VMM, but the I/O has not yet completed; or if the page was pre-fetched by VMM's read-ahead algorithm, but was hidden from the faulting segment; or if the page has been put on the free list and has not yet been reused

zero-filled page faults

Incremented if the page fault is to working storage and can be satisfied by assigning a frame and zero-filling it.

executable-filled page faults

Incremented for each instruction page fault.

pages examined by the clock

VMM uses a clock-algorithm to implement a pseudo least recently used (lru) page replacement scheme. Pages are aged by being examined by the clock. This count is incremented for each page examined by the clock.

revolutions of the clock hand

Incremented for each VMM clock revolution (that is, after each complete scan of memory).

pages freed by the clock

Incremented for each page the clock algorithm selects to free from real memory.

backtracks

Incremented for each page fault that occurs while resolving a previous page fault.

free frame waits

Incremented each time a process is waited by VMM while free frames are gathered.

extend XPT waits

Incremented each time a process is waited by VMM due to a commit in progress for the segment being accessed.

-s (continued)

pending I/O waits

Incremented each time a process is waited by VMM for a page-in I/O to complete.

start I/Os

Incremented for each read or write I/O request initiated by VMM. This count should equal the sum of page-ins and page-outs.

iodones

Incremented at the completion of each VMM I/O request.

CPU context switches

Incremented for each CPU context switch (dispatch of a new process).

device interrupts

Incremented on each hardware interrupt

software interrupts

Incremented on each software interrupt. A software interrupt is a machine instruction similar to a hardware interrupt that saves some state and branches to a service routine. System calls are implemented with software interrupt instructions that branch to the system call handler routine.

decrementer interrupts

Incremented on each decrementer interrupt.

mpc send interrupts

Incremented on each mpc send interrupt

mpc receive interrupts

Incremented on each mpc receive interrupt

phantom interrupts

Incremented on each phantom interrupt

traps Not maintained by the operating system.

syscalls

Incremented for each system call.

-I Displays I/O oriented view with the new columns of output, **p** under heading **kthr**, and columns **fi** and **fo** under heading **page** instead of the columns **re** and **cy** in the **page** heading.

-t Prints the time-stamp next to each line of output of **vmstat**. The time-stamp is displayed in the HH:MM:SS format.

Note: Time stamp will not be printed if **-f**, **-s**, or **-i** flags are specified.

-v Writes to standard output various statistics maintained by the Virtual Memory Manager. The **-v** flag can only be used with the **-s** flag. Possible values:

memory pages

Size of real memory in number of 4 KB pages.

lrutable pages

Number of 4 KB pages considered for replacement. This number excludes the pages used for VMM internal pages, and the pages used for the pinned part of the kernel text.

free pages

Number of free 4 KB pages.

memory pools

Tuning parameter (managed using **vmo**) specifying the number of memory pools.

pinned pages

Number of pinned 4 KB pages

maxpin percentage

Tuning parameter (managed using **vmo**) specifying the percentage of real memory which can be pinned.

minperm percentage

Tuning parameter (managed using **vmo**) in percentage of real memory. This specifies the point below which file pages are protected from the re-page algorithm.

maxperm percentage

Tuning parameter (managed using **vmo**) in percentage of real memory. This specifies the point above which the page stealing algorithm steals only file pages.

numperm percentage

Percentage of memory currently used by the file cache.

file pages

Number of 4 KB pages currently used by the file cache.

compressed percentage

Percentage of memory used by compressed pages.

	compressed pages	Number of compressed memory pages.
	numclient percentage	Percentage of memory occupied by client pages.
	maxclient percentage	Tuning parameter (managed using vmo) specifying the maximum percentage of memory which can be used for client pages.
-v (continued)	client pages	Number of client pages.
	remote pageouts scheduled	Number of pageouts scheduled for the client file systems.
	pending disk I/Os blocked with no pbuf	Number of pending disk I/O requests blocked because no pbuf was available. Pbufs are pinned memory buffers used to hold I/O requests at the logical volume manager layer.
	paging space I/Os blocked with no psbuf	Number of paging space I/O requests blocked because no psbuf was available. Psbufs are pinned memory buffers used to hold I/O requests at the virtual memory manager layer.
	filesystem I/Os blocked with no fsbuf	Number of file system I/O requests blocked because no fsbuf was available. Fsbuf are pinned memory buffers used to hold I/O requests in the file system layer.
	client filesystem I/Os blocked with no fsbuf	Number of client file system I/O requests blocked because no fsbuf was available. NFS (Network File System) and VxFS (Veritas) are client file systems. Fsbuf are pinned memory buffers used to hold I/O requests in the file system layer.
	external pager filesystem I/Os blocked with no fsbuf	Number of external pager client file system I/O requests blocked because no fsbuf was available. JFS2 is an external pager client file system. Fsbuf are pinned memory buffers used to hold I/O requests in the file system layer.
-w		Display the report in wide mode
-l		Displays an additional "large-page" section with the <i>alp</i> and <i>flp</i> columns.
-p <i>pagesize</i>		Appends the VMM statistics for the specified page size to the regular vmstat output.
-P <i>pagesize</i>		Displays only the VMM statistics which are relevant for the specified page size.
Interval		Specifies the amount of time in seconds between each report.
Count		Determines the number of reports generated and the number of seconds apart.

Exit Status

The following exit values are returned:

0	Successful completion.
>0	Invalid flag, argument, or command failure

Examples

1. To display a summary of the statistics since boot, type: :
vmstat
2. To display five summaries at 2-second intervals, type:
vmstat 2 5

3. To display a summary of the statistics since boot including statistics for logical disks `hdisk1` and `hdisk2`, type
`vmstat hdisk1 hdisk2`
4. To display fork statistics, type:
`vmstat -f`
5. To display the count of various events, type:
`vmstat -s`
6. To display time-stamp next to each column of output of **vmstat**, type:
`vmstat -t`
7. To display the I/O oriented view with an alternative set of columns, type:
`vmstat -I`
8. To display all the VMM statistics available, type:
`vmstat -v -s`
9. To display the large-page section with the *alp* and *flp* columns at 8-second intervals, type:
`vmstat -l 8`
10. To display the VMM statistics specific to a particular page size (in the example, 4K), type:
`vmstat -p 4K`
11. To display the VMM statistics for all page sizes that are supported on the system, type:
`vmstat -p ALL`

or
`vmstat -p all`
12. To display only the VMM statistics for a particular page size (in this example, 4K), type:
`vmstat -P 4K`
13. To display only the per-page breakdown of VMM statistics for all supported page sizes, type:
`vmstat -P ALL`

or
`vmstat -P all`

Related Information

The `cfgdev` command, the `chdev` command, the `chpath` command, the `lsdev` command, the `lsmmap` command, and the `rmdev` command.

wall command

Purpose

Writes a message to all users that are logged in.

Syntax

```
wall [ -a ] [ -g Group ][ Message ]
```

Description

The **wall** command writes a message to all users that are logged in. If the *Message* parameter is not specified, the **wall** command reads the message from standard input until it reaches an end-of-file character. The message is then sent to all logged in users. The following heading precedes the message:

Broadcast message from
user@node

(tty) at hh:mm:ss ...

hh:mm:ss represents the hours, minutes, and seconds when the message was sent.

To override any protections set up by other users, you must operate with root user authority. Typically, the root user uses the **wall** command to warn all other users of an impending system shut down.

Note:

1. The **wall** command only sends messages to the local node.
2. Messages can contain multibyte characters.

Flags

-a	Performs the default operation. This flag is provided for System V compatibility. It broadcast messages to the console and pseudo-terminals.
-g <i>Group</i>	Broadcasts to a specified group only.

Files

/dev/tty	Specifies a device.
----------	---------------------

wc command

Purpose

Counts the number of lines, words, bytes, or characters in a file.

Syntax

```
wc [ -c | -m ] [ -l ] [ -w ] [ File ... ]
```

```
wc -k [ -c ] [ -l ] [ -w ] [ File ... ]
```

Description

By default, the **wc** command counts the number of lines, words, and bytes in the files specified by the *File* parameter. The command writes the number of newline characters, words, and bytes to the standard output and keeps a total count for all named files.

When you use the *File* parameter, the **wc** command displays the file names as well as the requested counts. If you do not specify a file name for the *File* parameter, the **wc** command uses standard input.

The **wc** command is affected by the **LANG**, **LC_ALL**, **LC_CTYPE**, and **LC_MESSAGES** environment variables.

The **wc** command considers a word to be a string of characters of non-zero length which are delimited by a white space (for example SPACE , TAB).

Flags

- c** Counts bytes unless the **-k** flag is specified. If the **-k** flag is specified, the **wc** command counts characters.
- k** Counts characters. Specifying the **-k** flag is equivalent to specifying the **-klwc** flag. If you use the **-k** flag with other flags, then you must include the **-c** flag. Otherwise, the **-k** flag is ignored. For more information, see examples 4 and 5.
Note: This flag is to be withdrawn in a future release.
- l** Counts lines.
- m** Counts characters. This flag cannot be used with the **-c** flag.
- w** Counts words. A word is defined as a string of characters delimited by spaces, tabs, or newline characters.

Note: If no flag is specified, **wc** by default counts the lines, words, bytes in a file or from standard input.

Exit Status

This command returns the following exit values:

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

Examples

1. To display the line, word, and byte counts of a file, enter:

```
wc chap1
```

The **wc** command displays the number of lines, words, and bytes in the chap1 file.

2. To display only byte and word counts, enter:

```
wc -cw chap*
```

The **wc** command displays the number of bytes and words in each file that begins with chap. The command also displays the total number of bytes and words in these files.

3. To display the line, word, and character counts of a file, enter:

```
wc -k chap1
```

The **wc** command displays the number of lines, words, and characters in the chap1 file.

4. To display the word and character counts of a file, enter:

```
wc -kcw chap1
```

The **wc** command displays the number of characters and words in the chap1 file.

5. To use the **wc** command on standard input, enter:

```
wc -klw
```

The **wc** command displays the number of lines and words in standard input. The **-k** flag is ignored.

6. To display the character counts of a file, enter:

```
wc -m chap1
```

The **wc** command displays the number of characters in the chap1 file.

7. To use the **wc** command on standard input, enter:

```
wc -mlw
```

The **wc** command displays the number of lines, words, and characters in standard input.

Files

<code>/usr/bin/wc</code> , <code>/bin/wc</code>	Contains the wc command.
<code>/usr/ucb/wc</code>	Contains the symbolic link to the wc command.

who command

Purpose

Identifies the users currently logged in.

Syntax

```
who [ -a | -b -d -i -l -m -p -q -r -s -t -u -w -A -H -T -X ] [ File ]
```

```
who am { i | I }
```

Description

The **who** command displays information about all users currently on the local system. The following information is displayed: login name, tty, date and time of login. Typing `who am i` or `who am I` displays your login name, tty, date and time you logged in. If the user is logged in from a remote machine, then the host name of that machine is displayed as well.

The **who** command can also display the elapsed time since line activity occurred, the process ID of the command interpreter (shell), logins, logoffs, restarts, and changes to the system clock, as well as other processes generated by the initialization process.

The general output format of the **who** command is as follows:

```
Name [State] Line Time [Activity] [Pid] [Exit] (Hostname)
```

where:

Name	Identifies the user's login name.
State	Indicates whether the line is writable by everyone (see the -T flag).
Line	Identifies the line name as found in the /dev directory.
Time	Represents the time when the user logged in.
Activity	Represents the hours and minutes since activity last occurred on that user's line. A . (dot) here indicates line activity within the last minute. If the line has been quiet more than 24 hours or has not been used since the last system startup, the entry is marked as old.
Pid	Identifies the process ID of the user's login shell.
Term	Identifies the process termination status (see the -d flag).
Exit	Identifies the exit status of ended processes (see the -d flag).
Hostname	Indicates the name of the machine the user is logged in from.

To obtain information, the **who** command usually examines the **/etc/utmp** file. If you specify another file with the *File* parameter, the **who** command examines that file instead. This new file is usually the **/var/adm/wtmp** or **/etc/security/failedlogin** file.

If the *File* parameter specifies more than one file name, only the last file name will be used.

Note: This command only identifies users on the local node.

Flags

-a	Processes the /etc/utmp file or the named file with all information. Equivalent to specifying the -bdlpriTu flags.
-b	Indicates the most recent system startup time and date.
-d	Displays all processes that have expired without being regenerated by init . The exit field appears for dead processes and contains the termination and exit values (as returned by wait) of the dead process. (This flag is useful for determining why a process ended by looking at the error number returned by the application.)
-l	Lists any login process.
-m	Displays information about the current terminal only. The who -m command is equivalent to the who am i and who am I commands.
-p	Lists any active process that is currently active and has been previously generated by init .
-q	Prints a quick listing of users and the number of users on the local system.
-r	Indicates the current run-level of the process.
-s	Lists only the name, line, and time fields. This flag is the default; thus, the who and who -s commands are equivalent.
-t	Indicates the last change to the system clock by the root user using the date command. If the date command has not been run since system installation, the who -t command produces no output.
-u or -i	Displays the user name, tty, login time, line activity, and process ID of each current user.
-A	Displays all accounting entries in the /etc/utmp file.
-H	Displays a header (title).
-T or -w	Displays the state of the tty and indicates who can write to that tty as follows: + Writable by anyone. - Writable only by the root user or its owner. ? Bad line encountered.
-X	Prints all available characters of each user name instead of truncating to the first 8 characters. The user name is also moved to the last column of the output.

Exit Status

This command returns the following exit values:

0	Successful completion.
>0	An error occurred.

Examples

1. To display information about who is using the local system node, type:

```
who
```

Information similar to the following is displayed:

```
pts/1 Nov 9 00:20 long_username_greater_than_eight_characters (localhost)
```

2. To display your user name, type:

```
who am i
```

Information similar to the following is displayed:

```
george lft/0 Jun 8 08:34
```

3. To display a history of logins, logouts, system start ups, and system shut downs, type:

```
who /var/adm/wtmp
```

Information similar to the following is displayed:

```

hank  lft/0  Jun  8  08:34  (ausnix5)
john  lft/0  Jun  8  08:34  (JIKey)
mary  lft/0  Jun  8  08:22  (machine.austin.ibm)
jan   pts4    Jun  8  09:19  (puff.wisc.edu)

```

4. To display the run-level of the local system node, type:

```
who -r
```

Information similar to the following is displayed:

```
. run-level 2 Jun 8 04:15 2 0 s
```

5. To display any active process that is currently actively and has been previously generated by init, type:

```
who -p
```

Information similar to the following is displayed:

```
srcmstr . Jun 8 04:15 old 2896
cron . Jun 8 04:15 old 4809
uprintfd . Jun 8 04:15 old 5158
```

6. To process the `/var/adm/wtmp` file with the `-bdlprtTu` flags specified, type:

```
who -a/var/adm/wtmp
```

Information similar to the following is displayed:

```

.      system boot Jun 19 10:13
.      run-level 2 Jun 19 10:13
.      . Jun 19 10:14 old
.      . Jun 19 10:14 old
.      . Jun 19 10:14 old
rc     - . Jun 19 10:13 old
.      . Jun 19 10:16 old
.      . Jun 19 10:14 old
srcmstr - . Jun 19 10:14 old
rctcpip - . Jun 19 10:14 old
rcdce  - . Jun 19 10:14 old
rccm   - . Jun 19 10:15 old
dceupdt - . Jun 19 10:15 old
rcnfs  - . Jun 19 10:15 old
cron   - . Jun 19 10:16 old
piobe  - . Jun 19 10:16 old
qdaemon - . Jun 19 10:16 old
writesrv - . Jun 19 10:16 old
uprintfd - . Jun 19 10:16 old
.      . Jun 19 10:16 old
LOGIN  - lft0 Jun 19 10:16 old
.      . Jun 19 10:16 old
.      . Jun 19 10:16 old

```

Files

<code>/etc/utmp</code>	Contains user and accounting information.
<code>/etc/security/failedlogin</code>	Contains the history of all invalid logins.
<code>/var/adm/wtmp</code>	Contains the history of all logins since the file was last created.
<code>/usr/include/sys/signal.h</code>	Contains a list of termination values.

Related Information

The `date` command.

wkldagent command

Purpose

Starts, stops, or queries the state of the Workload Manager Agent.

Syntax

```
wkldagent -start | -status | -stop
```

Description

The **wkldagent** command starts, stops, and queries the state of the Workload Manager Agent. The Workload Manager Agent provides recording capability for a limited set of local system performance metrics. These include common CPU, memory, network, disk, and partition metrics typically displayed by the **topas** command.

The Workload Manager must be started using the **wkldmgr** command before the **wkldagent** command is run. Daily recordings are stored in the **/home/ios/perf/wlm** directory with filenames **xmwlm.YYMMDD**, where **YY** is the year, **MM** is the month, and **DD** is the day. The **wkldout** command can be used to process Workload Manager-related recordings. All recordings cover 24-hour periods and are retained for only two days.

Flags

-start	Starts the Workload Manager Agent.
-status	Displays the state of the Workload Manager Agent, either running or stopped.
-stop	Stops the Workload Manager Agent.

Exit Status

0	The command completed successfully
>0	An error occurred.

Examples

1. To start the Workload Manager Agent, type:

```
wkldmgr -start
```
2. To check whether the Workload Manager Agent is currently active, type:

```
wkldmgr -status
```
3. To stop the Workload Manager Agent, type:

```
wkldmgr -stop
```

Related Information

The **topas** command, the **wkldmgr** command, and the **wkldout** command.

wkldmgr command

Purpose

Starts or stops Workload Manager.

Syntax

`wkldmgr -start | -status | -stop`

Description

The `wkldmgr` command starts, stops, and queries the state of the Workload Manager. Starting the Workload Manager is necessary for the `-cecdisp` option of the `topas` command to work properly.

Flags

<code>-start</code>	Starts the Workload Manager.
<code>-status</code>	Displays the state of the Workload Manager, either running or stopped.
<code>-stop</code>	Stops the Workload Manager.

Exit Status

0	The command completed successfully
>0	An error occurred.

Examples

1. To start the Workload Manager, type:
`wkldmgr -start`
2. To check whether the Workload Manager is currently active, type:
`wkldmgr -status`
3. To stop the Workload Manager, type:
`wkldmgr -stop`

Related Information

The `topas` command, the `wkldagent` command, and the `wkldout` command.

wkldout command

Purpose

Provides post-processing of the recordings made by the Workload Manager Agent (`wkldagent`).

Syntax

```
wkldout [-report reportType] [ -interval MM ][-beg HHMM][ -end HHMM] [ -fmt [-mode modeType ] ] [ -graph ] -filename <xmwlm_recording_file>
```

Description

The `wkldout` command provides post-processing of recordings made by `wkldagent`. It can generate reports in ASCII or spreadsheet format. The output can be formatted by using the flags detailed, summary, disk or LAN. The time interval can be specified to process the `xmwlm` file for a particular interval.

Flags

-report <i>report</i> < <i>type</i> >	Generates the report based on the type specified. The report type can be detailed, summary, LAN, or disk. The detailed report type produces the comprehensive output of the <i>xmwl</i> m file and displays the report on the screen. The summary report type produces the abstract output of the <i>xmwl</i> m file and displays on the report on the screen. The LAN report type generates the summary output related to system LAN from the given <i>xmwl</i> m file and displays on the report on the screen. The disk report type generates I/O summary information and generates the <i>xmwl</i> m file and displays on the report on the screen.
-interval <i>MM</i>	Specifies how the recording reports are split into equal size time periods. Allowed Values (in minutes) are 5, 10, 15, 30, 60. If <i>-i</i> flag is not specified, a 5 minute interval is taken by default.
-beg <i>HHMM</i>	Indicates the beginning time in hours (HH) and minutes (MM). The range is between 0000 and 2400.
-end <i>HHMM</i>	Indicates the ending time in hours (HH) and minutes (MM). The range is between 0000 and 2400 and is greater than the begin time.
-fmt	Generates a spreadsheet format output in a file under <i>/home/ios/perf/wlm</i> in the format <i>xmwl</i> m.YYMMDD_01.
-mode < <i>type</i> >	Specifies the post-processor only output mean values by default. Other values and the full set are available via other options (min, max, mean, stdev, set).
-graph	Generate the csv file under <i>/home/ios/perf/wlm</i> in the format <i>xmwl</i> m.YYMMDD.csv, which can be input to Nmon analyzer to produce graphs to aid in analysis and writing reports. The Nmon analyzer requires Excel 2002 or later.
-filename < <i>name</i> >	Name of the wkldagent recording file. These files will be generated by the wkldagent command and resides in <i>/home/ios/perf/wlm</i> in the format <i>xmwl</i> m.YYMMDD

Exit Status

0	The command completed successfully
>0	An error occurred.

Examples

1. To display the system summary from *xmwl*m.070731, type the following command :
`wkldout -report summary -filename /home/ios/perf/wlm/xmwl`m.070731
2. To display the detailed report of the system from *xmwl*m.070731, type the following command:
`wkldout -report detailed -filename /home/ios/perf/wlm/xmwl`m.070731
3. To display the disk I/O summary from *xmwl*m.070731, type the following command:
`wkldout -report disk -filename /home/ios/perf/wlm/xmwl`m.070731
4. To display the system lan summary from *xmwl*m.070731, type the following command:
`wkldout -report lan -filename /home/ios/perf/wlm/xmwl`m.070731
5. To produce the csv file from *xmwl*m.070731, type the following command:
`wkldout -graph -filename /home/ios/perf/wlm/xmwl`m.070731
6. To display the system summary from *xmwl*m.070731 with time interval 10 min, type the following command:
`wkldout -report summary -interval 10 -filename /home/ios/perf/wlm/xmwl`m.070731
7. To display the detailed report from *xmwl*m.070731, starting at 01:00 Hr and ending 02:00 Hr, type
`wkldout -report detailed -beg 0100 -end 0200`
`-filename /home/ios/perf/wlm/xmwl`m.070731

Related Information

The **topas** command, the **wkldmgr** command, and the **wkldagent** command.

Appendix. Notices

This information was developed for products and services offered in the U.S.A.

The manufacturer may not offer the products, services, or features discussed in this document in other countries. Consult the manufacturer's representative for information on the products and services currently available in your area. Any reference to the manufacturer's product, program, or service is not intended to state or imply that only that product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any intellectual property right of the manufacturer may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any product, program, or service.

The manufacturer may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to the manufacturer.

For license inquiries regarding double-byte character set (DBCS) information, contact the Intellectual Property Department in your country or send inquiries, in writing, to the manufacturer.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: THIS INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. The manufacturer may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to Web sites not owned by the manufacturer are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this product and use of those Web sites is at your own risk.

The manufacturer may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact the manufacturer.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, IBM License Agreement for Machine Code, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have

been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning products not produced by this manufacturer was obtained from the suppliers of those products, their published announcements or other publicly available sources. This manufacturer has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to products not produced by this manufacturer. Questions on the capabilities of products not produced by this manufacturer should be addressed to the suppliers of those products.

All statements regarding the manufacturer's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

The manufacturer's prices shown are the manufacturer's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to the manufacturer, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. The manufacturer, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information in softcopy, the photographs and color illustrations may not appear.

Programming interface information

This (ADD NAME OF PUBLICATION HERE) publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of (ADD PRODUCT NAME HERE).

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at www.ibm.com/legal/copytrade.shtml.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Red Hat, the Red Hat "Shadow Man" logo, and all Red Hat-based trademarks and logos are trademarks or registered trademarks of Red Hat, Inc., in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Terms and conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

Personal Use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these publications, or any portion thereof, without the express consent of the manufacturer.

Commercial Use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of the manufacturer.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any data, software or other intellectual property contained therein.

The manufacturer reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by the manufacturer, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

THE MANUFACTURER MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THESE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

BULL CEDOC
357 AVENUE PATTON
B.P.20845
49008 ANGERS CEDEX 01
FRANCE

REFERENCE
86 A1 41EV 05