

Chapter 8

Filtering Command Output

For commands that display output, such as the `show` commands, you can filter the output. This chapter provides information about the following topics:

- Using Keyboard Sequences at the `---(more)---` Prompt on page 137
- Using the Pipe (`|`) Symbol When Entering Commands on page 139

Using Keyboard Sequences at the `---(more)---` Prompt

If the output from a command is longer than the screen length, it appears one screen at a time by means of a UNIX `more`-type interface. The prompt `---(more)---` indicates that more output is available. This format is helpful when you want to scroll and search through lengthy output.

As soon as the command-line interface (CLI) can determine the length of the output (typically by the second screen), it displays the percentage of the output that has been displayed so far above the `---(more)---` prompt.

The output buffer for the prompt is restricted to 32 megabytes (MB). Any new data that exceeds the buffer limit replaces the oldest data in the memory buffer. When the buffer limit is exceeded, attempts to search backward or navigate to the beginning of the output generate a warning indicating that the output is truncated.

Because of the buffer size restriction, use of the scroll and search functions might be limited.

Table 15 lists the keyboard sequences you can use at the `—(more)—` prompt.

Table 15: `—(more)—` Prompt Keyboard Sequences

Category	Action	Keyboard Sequence
Get Help	Display information about the keyboard sequences you can display at the <code>—(more)—</code> prompt.	h
Scroll Down	Scroll down one line.	Enter, k, Ctrl+m, Ctrl+n, or down arrow
	Scroll down one-half screen.	Tab, d, Ctrl+d, or Ctrl+x
	Scroll down one whole screen.	Space or Ctrl+f
	Scroll down to the bottom of the output.	Ctrl+e or G
	Display the output all at once instead of one screen at a time. (Same as specifying the <code>no-more</code> filter when entering commands using the pipe symbol. See “Preventing Output from Being Paginated” on page 144.)	N
Scroll Up	Display the previous line of output.	j, Ctrl+h, Ctrl+p, or up arrow
	Scroll up one-half screen.	u or Ctrl+u
	Scroll up one whole screen.	b or Ctrl+b
	Scroll up to the top of the output.	Ctrl+a or g
Search	Search forward for a string.	/string
	Search backward for a string.	?string
	Repeat the previous search for a string.	n
	Search for a text string. You are prompted for the string to match. (Same as specifying the <code>match</code> filter when entering command using the pipe symbol. See “Displaying Output That Matches a Regular Expression” on page 144.)	m or M
	Search, ignoring a text string. You are prompted for the string to not match. (Same as specifying the <code>except</code> filter when entering command using the pipe symbol. See “Ignoring Output That Does Not Match a Regular Expression” on page 143.)	e or E
Interrupt or End Output, Redraw the Output, and Save the Output to a File	Interrupt the display of output.	Ctrl+c, q, Q, or Ctrl+k
	Do not redisplay the CLI prompt immediately after displaying the output, but remain at the <code>—(more)—</code> prompt. (Same as specifying the <code>hold</code> command.)	H
	Clear any match conditions and display the complete output.	c or C
	Redraw the output on the screen.	Ctrl+l
	Save the command output to a file. You are prompted for a filename. (Same as specifying the <code>save filename</code> command.)	s or S

Using the Pipe (|) Symbol When Entering Commands

You can filter output by adding the | (or *pipe*) symbol when you enter a command. For example:

```
user@host> show rip neighbor ?
Possible completions:
<[Enter]>      Execute this command
<name>        Name of RIP neighbor
instance      Name of RIP instance
logical-router Name of logical router, or 'all'
|             Pipe through a command
```

The following example lists the filters that can be used with the pipe symbol:

```
user@host> show rip neighbor | ?
Possible completions:
count          Count occurrences
display        Show additional kinds of information
except         Show only text that does not match a pattern
find           Search for first occurrence of pattern
hold           Hold text without exiting the --More-- prompt
last           Display end of output only
match          Show only text that matches a pattern
no-more        Don't paginate output
request        Make system-level requests
resolve        Resolve IP addresses
save           Save output text to file
trim           Trim specified number of columns from start of line
```

For the show configuration command only, an additional compare filter is available:

```
user@host> show configuration | ?
Possible completions:
compare        Compare configuration changes with prior version
...
```

You can enter any of the pipe filters in conjunction. For example:

```
user@host> command | match regular-expression | save filename
```

See “Pipe Filter Functions” on page 141 for a description of each type of filter.



NOTE: This section describes *only* the filters that can be used for operational mode command output. For information about filters that can be used in configuration mode, see the *JUNOS System Basics Configuration Guide*.

Using Regular Expressions with the Pipe Symbol

The `except`, `find`, and `match` filters used with the pipe symbol employ regular expressions to filter output. Juniper Networks uses the regular expressions as defined in POSIX 1003.2. (See Table 16.) If the regular expression contains spaces, operators, or wildcard characters, enclose the expression in quotation marks.

Table 16: Common Regular Expression Operators in Operational Mode Commands

Operator	Function
	Indicates that a match can be one of the two terms on either side of the pipe.
^	Used at the beginning of an expression, denotes where a match should begin.
\$	Used at the end of an expression, denotes that a term must be matched exactly up to the point of the \$ character.
[]	Specifies a range of letters or digits to match. To separate the start and end of a range, use a hyphen (-).
()	Specifies a group of terms to match.

For example, if a command produces the following output:

```
1 2
2 2
3 2 1
4
```

a pipe filter of `| match 2` displays the following output:

```
1 2
2 2
3 2 1
```

and a pipe filter of `| except 1` displays the following output:

```
2 2
4
```



NOTE: See the following sections for more examples of using regular expressions:

- Ignoring Output That Does Not Match a Regular Expression on page 143
- Displaying Output from the First Match of a Regular Expression on page 143
- Displaying Output That Matches a Regular Expression on page 144

Pipe Filter Functions

This section describes each pipe filter:

- Comparing Configurations on page 141
- Counting the Number of Lines of Output on page 142
- Displaying Output in XML Tag Format on page 142
- Ignoring Output That Does Not Match a Regular Expression on page 143
- Displaying Output from the First Match of a Regular Expression on page 143
- Retaining Output After the Last Screen on page 143
- Displaying Output Beginning with the Last Entries on page 144
- Displaying Output That Matches a Regular Expression on page 144
- Preventing Output from Being Paginated on page 144
- Sending Command Output to Other Users on page 144
- Resolving IP Addresses on page 145
- Saving Output to a File on page 145
- Trimming Output by Specifying the Starting Column on page 145

Comparing Configurations

The `compare` filter compares the candidate configuration with either the current committed configuration or a configuration file and displays the differences between the two configurations. To compare configurations, enter `compare` after the pipe symbol:

```
[edit]
user@host# show | compare [filename | rollback n]
```

filename is the full path to a configuration file.

n is the index into the list of previously committed configurations. The most recently saved configuration is 0. If you do not specify arguments, the candidate configuration is compared against the active configuration file (`/config/juniper.conf`).

The comparison output uses the following conventions:

- Statements that are only in the candidate configuration are prefixed with a plus sign (+).
- Statements that are only in the comparison file are prefixed with a minus sign (-).
- Statements that are unchanged are prefixed with a single blank space ().

For example:

```

user@host> show configuration system | compare rollback 9
[edit system]
+ host-name nutmeg;
+ backup-router 192.168.71.254;
- ports {
-   console log-out-on-disconnect;
- }
[edit system name-server]
+ 172.17.28.11;
  172.17.28.101 { ... }
[edit system name-server]
  172.17.28.101 { ... }
+ 172.17.28.100;
+ 172.17.28.10;
[edit system]
- scripts {
-   commit {
-     allow-transients;
-   }
- }
+ services {
+   ftp;
+   rlogin;
+   rsh;
+   telnet;
+ }

```

Counting the Number of Lines of Output

To count the number of lines in the output from a command, enter `count` after the pipe symbol. For example:

```

user@host> show configuration | count
Count: 269 lines

```

Displaying Output in XML Tag Format

To display command output in XML tag format, enter `display xml` after the pipe symbol.

The following example displays the `show cli directory` command output as XML tags:

```

user@host> show cli directory | display xml
<rpc-reply xmlns:junos="http://xml.juniper.net/junos/7.5I0/junos">
  <cli>
    <working-directory>/var/home/regress</working-directory>
  </cli>
  <cli>
    <banner></banner>
  </cli>
</rpc-reply>

```

Ignoring Output That Does Not Match a Regular Expression

To ignore text that matches a regular expression, specify the `except` command after the pipe symbol. If the regular expression contains any spaces, operators, or wildcard characters, enclose it in quotation marks. For information on common regular expression operators, see Table 16 on page 140.

The following example displays all users who are logged in to the router, except for the user `root`:

```
user@host> show system users | except root
      8:28PM up 1 day, 13:59, 2 users, load averages: 0.01, 0.01, 0.00
USER   TTY FROM                LOGIN@  IDLE WHAT
sheep  p0  baa.juniper.net      7:25PM   - cli
```

Displaying Output from the First Match of a Regular Expression

To display output starting with the first occurrence of text matching a regular expression, enter `find` after the pipe symbol. If the regular expression contains any spaces, operators, or wildcard characters, enclose it in quotation marks. For information on common regular expression operators, see Table 16 on page 140.

The following example displays the routes in the routing table starting at IP address `208.197.169.0`:

```
user@host> show route | find 208.197.169.0
208.197.169.0/24  *[Static/5] 1d 13:22:11
                  > to 192.168.4.254 via so-3/0/0.0
224.0.0.5/32     *[OSPF/10] 1d 13:22:12, metric 1

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

47.0005.80ff.f800.0000.0108.0001.1921.6800.4015.00/160
                  *[Direct/0] 1d 13:22:12
                  > via lo0.0
```

The following example displays the first CCC entry in the forwarding table:

```
user@host> show route forwarding-table | find ccc
Routing table: ccc
MPLS:
Interface.Label  Type RtRef Nexthop          Type Index NhRef Netif
default         perm  0          10.0.16.2          rjct  3    1
0               user  0          10.0.16.2          recv  5    2
1               user  0          10.0.16.2          recv  5    2
32769           user  0          10.0.16.2          ucst  45   1 fe-0/0/0.534
fe-0/0/0. (CCC) user  0          10.0.16.2          indr  44   2
                                     Push 32768, Push
```

Retaining Output After the Last Screen

To not return immediately to the CLI prompt after viewing the last screen of output, enter `hold` after the pipe symbol. The following example prevents returning to the CLI prompt after you have viewed the last screen of output from the `show log log-file-1` command:

```
user@host> show log log-file-1 | hold
```

This filter is useful when you want to scroll or search through output.

Displaying Output Beginning with the Last Entries

To display text starting from the end of the output, enter `last <lines>` after the pipe symbol.

The following example displays the last entries in `log-file-1` file:

```
user@host> show log log-file-1 | last
```

This filter is useful for viewing log files in which the end of the file contains the most recent entries.

Displaying Output That Matches a Regular Expression

To display output that matches a regular expression, enter `match regular-expression` after the pipe symbol. If the regular expression contains any spaces, operators, or wildcard characters, enclose it in quotation marks. For information on common regular expression operators, see Table 16 on page 140.

The following example matches all the Asynchronous Transfer Mode (ATM) interfaces in the configuration:

```
user@host> show configuration | match at-
  at-2/1/0 {
  at-2/1/1 {
  at-2/2/0 {
  at-5/2/0 {
  at-5/3/0 {
```

Preventing Output from Being Paginated

By default, if output is longer than the length of the terminal screen, you are provided with a `-(more)-` message to display the remaining output. To display the remaining output, press the Spacebar.

To prevent the output from being paginated, enter `no-more` after the pipe symbol.

The following example displays output from the `show configuration` command all at once:

```
user@host> show configuration | no-more
```

This feature is useful, for example, if you want to copy the entire output and paste it into an e-mail.

Sending Command Output to Other Users

To display command output on the terminal of a specific user logged in to your router, or on the terminals of all users logged in to your router, enter `request message (all | user account@terminal)` after the pipe symbol.

If you are troubleshooting your router and, for example, talking with a customer service representative on the phone, you can use the `request message` command to send your representative the command output you are currently viewing on your terminal.

The following example sends the output from the `show interfaces` command you enter on your terminal to the terminal of the user `root@tty1`:

```
user@host> show interfaces | request message user root@tty1
```

The user `root@tty1` sees the following output appear on the terminal screen:

```
Message from user@host on /dev/tty0 at 10:32 PST...
Physical interface: dsc, Enabled, Physical link is Up
Interface index: 5, SNMP ifIndex: 5
Type: Software-Pseudo, MTU: Unlimited...
```

Resolving IP Addresses

If the output of a command displays an unresolved IP address, you can enter `| resolve` after the command to display the name associated with the IP address. The `resolve` filter enables the system to perform a reverse DNS lookup of the IP address. If DNS is not enabled, the lookup fails and no substitution is performed.

To perform a reverse DNS lookup of an unresolved IP address, enter `resolve <full-names>` after the pipe symbol. If you do not specify the `full-names` option, the name is truncated to fit whatever field width limitations apply to the IP address.

The following example performs a DNS lookup on any unresolved IP addresses in the output from the `show ospf neighbors` command:

```
user@host> show ospf neighbors | resolve
```

Saving Output to a File

When command output is lengthy, when you need to store or analyze the output, or when you need to send the output in an e-mail or by FTP, you can save the output to a file. By default, the file is placed in your home directory on the router.

To save command output to a file, enter `save filename` after the pipe symbol.

The following example saves the output from the `request support information` command to a file named `my-support-info.txt`:

```
user@host> request support information | save my-support-info.txt
Wrote 1143 lines of output to 'my-support-info.txt'
user@host>
```

Trimming Output by Specifying the Starting Column

Output appears on the terminal screen in terms of rows and columns. The first alphanumeric character starting at the left of the screen is in column 1, the second character is in column 2, and so on. To display output starting from a specific column (thus trimming the leftmost portion of the output), enter `trim columns` after the pipe symbol. The `trim` filter is useful for trimming the date and time from the beginning of system log messages.

The following example displays output from the `show system storage` command, filtering out the first 10 columns:

```
user@host> show system storage | trim 11
```

