NORTEL NETWORKS

Preside Multiservice Data Manager

EPIC

Reference Guide

241-6001-809

Preside Multiservice Data Manager

EPIC

Reference Guide

Publication: 241-6001-809
Document status: Standard
Document version: 15.1RSUP
Document date: August 2004

Printed in Canada

Publication history

August 2004

15.1 RSUP Standard
Commercial availability except for MPE support which will be available in a future release.

Contents

About this document

This document describes how to use the Enhanced Passport Interface Controller (EPIC) command line interface to perform surveillance and provisioning tasks on Passport devices.

The following topics are discussed in this section:

Who should read this document and why

This document is intended for personnel who use EPIC to manage Passport devices.

What you need to know

Users of this document require the following knowledge and skills:

- working knowledge of UNIX and the Solaris operating environment
- knowledge of the Passport network and service data
- knowledge of Passport's component administrative system (CAS).

How this document is organized

241-6001-809 *Preside MDM EPIC Reference Guide* contains the following sections:

- "Introducing the EPIC command line interface" (page 15) gives a brief overview of EPIC and its functionality.
- "EPIC files" (page 17) has information about the special files that come with EPIC.
- "Running EPIC" (page 27) contains general information on using EPIC.
- "EPIC commands" (page 33) contains detailed information, including syntax, on EPIC commands.
- "EPIC special characters" (page 53) contains information on EPIC functions that require special characters in the command.

What's new in this document

There have not been any new changes to this NTP for this release.

Text conventions

This document uses the following text conventions:

- `nonproportional spaced plain type`

  Nonproportional spaced plain type represents system generated text or text that appears on your screen.

- **nonproportional spaced bold type**

  Nonproportional spaced bold type represents words that you should type or that you should select on the screen.

- *italics*

  Statements that appear in italics in a procedure explain the results of a particular step and appear immediately following the step.

  Words that appear in italics in text are for naming.

- [optional_parameter]

  Words in square brackets represent optional parameters. The command can be entered with or without the words in the square brackets.

- <general_term>

  Words in angle brackets represent variables which are to be replaced with specific values.

- UPPERCASE,lowercase

  In Preside Multiservice Data Manager (MDM), uppercase and lowercase letters that appear in UNIX commands and parameters must be matched exactly. The system matches upper and lowercase characters differently.

- UPPERCASE,lowercase

  Passport commands are not case-sensitive and do not have to match commands and parameters exactly as shown in this document, with the exception of string options values (for example, file and directory names) and string attribute values.

- |

  This symbol separates items from which you may select one; for example, ON|OFF indicates that you may specify ON or OFF. If you do not make a choice, a default ON is assumed.

- ...

  Three dots in a command indicate that the parameter may be repeated more than once in succession.

The term absolute pathname refers to the full specification of a path starting from the root directory. Absolute pathnames always begin with the slash (/) symbol. A relative pathname takes the current directory as its starting point, and starts with any alphanumeric character (other than /).

Related documents

See the following document for related information:

- NN10600-050 *Nortel Networks Multiservice Switch 7400/15000/20000 Command Reference*

Chapter 1
Introducing the EPIC command line interface

This section gives an overview of the Enhanced Passport Interface Controller (EPIC) command line interface. See the following sections for more information:

- "About EPIC" (page 15)
- "Configuring EPIC" (page 16)
- "Using EPIC" (page 16)

About EPIC

EPIC is an interactive command line interface that extends Passport's component administrative system (CAS). EPIC simplifies Passport surveillance and provisioning by providing the following functionality:

- simultaneous connection to multiple Passport switches
- extension of Passport's CAS syntax to include ranges and wildcards
- automation of common provisioning tasks
- enhanced display commands with results formatted in tables
- near real time monitoring of Passport components

For more information, see the following:

- "EPIC files" (page 17)
- "Running EPIC" (page 27)

- “EPIC commands” (page 33)
- “EPIC special characters” (page 53)

For more information on the Passport CAS, see NN10600-050 *Nortel Networks Multiservice Switch 7400/15000/20000 Command Reference*.

Configuring EPIC

Once you install EPIC, you can customize it through my modifying configuration files. See “EPIC files” (page 17).

Using EPIC

You can access EPIC from Preside Multiservice Data Manager (MDM) or from a command line.

Once you are in an EPIC session, you can use its commands and special characters to perform provisioning and surveillance tasks.

For more information, see the following:

- “Accessing EPIC” (page 27)
- “EPIC commands” (page 33)
- “EPIC special characters” (page 53)

Chapter 2
EPIC files

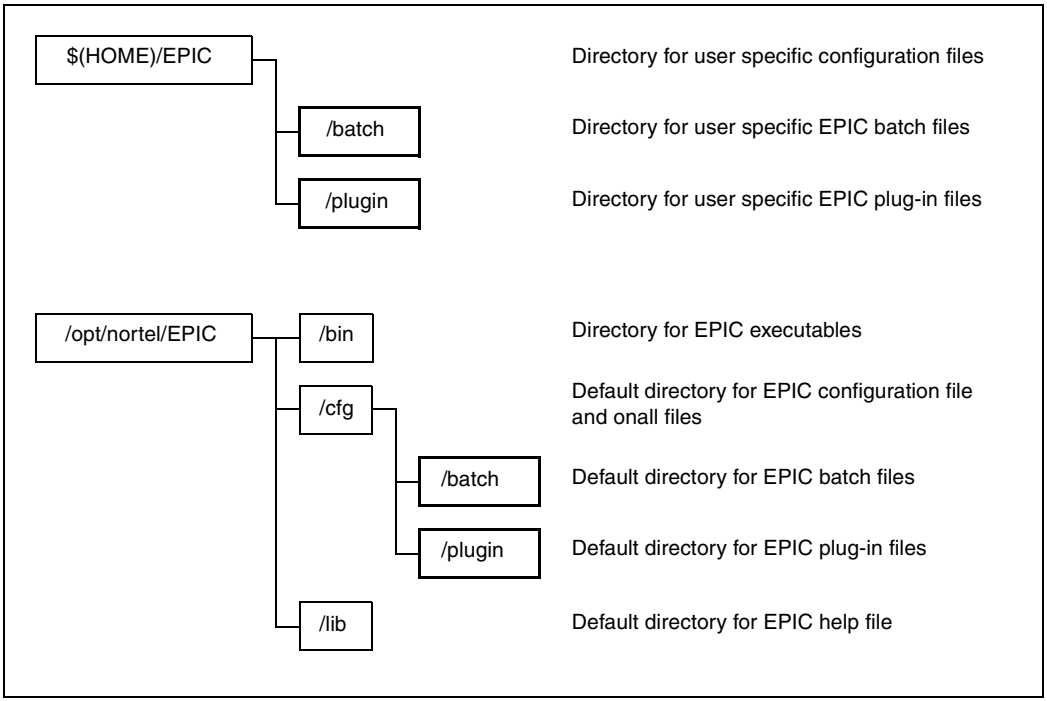
Enhanced Passport Interface Controller (EPIC) has several special files that you can copy and edit to customize your environment. The figure "EPIC directory structure" (page 18) shows the location of the EPIC files.

Note: Do not edit the default files in the /opt/nortel/EPIC directories.

See the following sections for information on the files that EPIC uses:

- "Configuration file" (page 18)
- "Batch files" (page 21)
- "Plug-in files" (page 22)
- ".netrc file" (page 24)
- "Log file" (page 24)
- "Onall file" (page 24)
- "Poll file" (page 25)

Figure 1
EPIC directory structure



Configuration file

The configuration file contains startup settings and login information. The filename is epic.config and EPIC searches for it in the following directories and in the following order:

- directory specified in command line option *-p <epic path>* if you are accessing EPIC from a command line
- subdirectory EPIC in your home directory
- /opt/nortel/EPIC/cfg

The following record types are used in the configuration file. If a record is omitted, the indicated default value for the parameter is used.

`default <userid> <password>` defines default login information for any switches not individually defined in the configuration file.

`<host> <userid> <password> [<ip address>]` defines login information for a particular switch. If the IP address is not specified, the switch must be defined in file /etc/hosts or through DNS.

`alarms [on|off|hidden]` defines whether alarms are enabled, disabled, or buffered but not displayed. Default is off.

`autocommit [on|off]` defines whether EPIC automatically saves and commits when you activate provisioning. Default is off.

`autoconfirm [on|off]` defines whether EPIC automatically confirms when you activate provisioning. Default is on.

`autosave [on|off]` defines whether EPIC automatically saves when you activate provisioning. Default is off.

`beeper [on|off]` turns the beeper on and off. Default is on.

`highlight [on|off]` sets the highlight mode. When on, values that have been changed between DISPLAY commands are highlighted. Default is on.

`logsize <alarm buffer size>` sets the size of the alarm buffer. Default is 100 alarms.

`maxprompt <number of switches>` sets the maximum number of switches to display in the command line prompt. Default is 10.

`multiswitch [on|off]` sets the multiswitch mode. When off, EPIC can connect to only one switch at a time. Default is on.

`pager [on|off]` sets the pagination. When on, EPIC prompts to continue after each full page of output. Default is on.

`pluginseq [on|off]` when set on, causes plug-ins to be run sequentially on the switches. When off, plug-ins are run simultaneously on the switches. Default is on.

`tables [on|off]` formats the output from DISPLAY commands into tables. Default is on.

`wildcard [smart|all]` when set to smart, ignores EPIC's wildcards, such as **, for LS and DISPLAY commands. Default is smart.

`alias <from>=<to>` defines an alias. The <from> value is replaced by the <to> value in EPIC commands.

Example

```
# Sample configuration file
#
default myuser mypassword

alarms off
autocommit off
autosave off
beeper on
highlight on
logsize 100
maxprompt 10
multiswitch on
pager on
pluginseq on
tables on
wildcard smart

labswitch-1 me blah
labswitch-2 me blah 42.1.2.3
```

In the example the default userid and password are myuser and mypassword. This login information is used for all switches except for labswitch-1 and labswitch-2.

You can change any setting in the configuration file by issuing the corresponding command during your EPIC session. For example, you can change the default login information with the USER command. See "EPIC commands" (page 33) for more information.

Batch files

You can automate common tasks by using batch files. Batch files contain lists of EPIC commands with user-supplied variables.

EPIC searches for batch files in the following directories and in the following order:

- subdirectory *batch* specified in command line option *-p <epic path>* if you are accessing EPIC from a command line
- subdirectory EPIC/batch in your home directory
- /opt/nortel/EPIC/cfg/batch

Guidelines for creating and storing batch files

- You must include the file extension .batch in the name of a batch file. If you omit the extension, you will not be able to run the batch file unless you specify the full path name of the batch file.
- Place batch files for automatic tasks common to all users in opt/nortel/EPIC/cfg/batch.
- Subdirectory opt/nortel/EPIC/cfg/batch includes a set of batch files that can be used as templates. Do not modify these template files. Instead copy them, rename them, and modify the renamed copy.
- Place batch files that apply one to one user in subdirectory EPIC/batch in the user's home directory.
- Do not store anything other than batch files in opt/nortel/EPIC/cfg/batch and /home/EPIC/batch. If you call a file in these directories and the file is not a valid batch file, the EPIC software will attempt to run the file anyway, with unexpected results.

  Note: A batch file may not function correctly with every Passport and release and should be verified for your network.

Types of batch files

The most common types of batch files contain EPIC commands to be executed. In addition files to that contain EPIC commands, there are two other types of batch files, batch data files and batch output files.

- Batch data files contain values for the variables in a batch file that contains EPIC commands.
- Batch output files contain responses to commands in file containing a batch of commands. You can redirect the responses to EPIC commands to a batch output file.

For more information on executing batch files and their syntax, see "Running batch files" (page 29).

Plug-in files

You can write plug-in scripts in any interpreted script language, such as c-shell or Perl. This feature allows for the use of intelligent scripts that can branch, based on command results.

EPIC cannot run a plug-in file until it is executable. Make sure that you run a chmod +x command on the file after it has been created or edited.

EPIC searches for plug-in files in the following directories and in the following order:

- subdirectory *plugin* specified in command line option *-p <epic path>* if you are accessing EPIC from a command line
- subdirectory EPIC/plugin in your home directory
- /opt/nortel/EPIC/cfg/plugin

  Note: Plug-in files should be placed in /opt/nortel/EPIC/cfg/plugin to automate common tasks. Sample plug-in files are included in this directory for use as templates. These plug-in files may not function correctly with every Passport release and should be verified for your network.

The following commands control EPIC from within a plug-in file:

- `epicSend` sends a command to EPIC and waits for it to complete.

- `epicNoEcho` hides Passport responses.
- `epicEcho` shows Passport responses.

When plug-ins run, they store temporary files in the directory /tmp. Results from each EPIC command are stored in a file whose name is stored in the $RESULTFILE environment variable. The result file is overwritten every time a new EPIC command is issued. As a plug-in file is run, the current switch name is stored in the $EPICSWITCH environment variable.

C-shell example

```
#!/bin/csh

echo "What release do you want to upgrade to (e.g.
BD02S9D)"
set release = $<

if ("$release" != "") then
epicNoEcho
epicSend "start prov"
grep -s "ok " $RESULTFILE
if ($status != 0) then
echo "Failed to start provisioning mode"
exit
endif

epicSend "d sw"
set newav1 = 'awk 'NR>1 {print substr($0,14,80)}'
$RESULTFILE
|grep -v ":"|sed 's/,/ /g'
|awk '{print $1"\n"$2"\n"$3"\n"$4"\n"$5"\n"}'
|awk 'NF>0' |sed 's/_.*/_'$release'/g''

epicSend "set sw avlist ! $newavl"
epicEcho

echo "Software changed to:"
echo $newavl
endif
```

Perl example

```
#!/bin/perl

syswrite(STDOUT,"What release do you want to upgrade to
(e.g. BD02S9D)\n");

$release=<STDIN>;

if ( $release ne "") {
   system("/opt/nortel/EPIC/bin/epicSend start prov");
   # get environment variables
   $resultFile=$ENV{'RESULTFILE'};
   $epicSwitch = $ENV{'EPICSWITCH'};
   syswrite(STDOUT,"result file: $resultFile\n");
   syswrite(STDOUT,"Switch: $epicSwitch\n");
   system("/opt/nortel/EPIC/bin/epicSend d sw");
}
```

.netrc file

The .netrc file, which is in your home directory, contains login information for FTP connections to remote hosts. It is also used by EPIC's FTP command. See "FTP" (page 40).

For more information on the .netrc file, see the UNIX man pages.

Log file

EPIC can log its session to a user specified log file. See "LOG" (page 41).

Onall file

The onall file contains a list of switch names, one per line. When EPIC's ONALL command is executed, connections to all the switches in the specified file are opened and the specified command is run simultaneously on every switch. See "ONALL" (page 42).

EPIC searches for onall files in the following directories and in the following order:

- directory specified in command line option *-p <epic path>* if you are accessing EPIC from a command line

- subdirectory EPIC in your home directory
- /opt/nortel/EPIC/cfg

Poll file

EPIC's POLL command directs output to a user specified poll file. The current date is appended to the poll filename. When a date change occurs, the current file is closed and a new file is opened. See "POLL" (page 43).

Chapter 3
Running EPIC

This section contains general information on using the Enhanced Passport Interface Controller (EPIC). See the following sections for more information:

- “Accessing EPIC” (page 27)
- “EPIC session features” (page 28)
- “Running batch files” (page 29)

Accessing EPIC

You can access EPIC from Preside Multiservice Data Manager (MDM) or from a command line. See the following sections:

- “Accessing EPIC from MDM” (page 27)
- “Accessing EPIC from a command line” (page 27)

Accessing EPIC from MDM

From the System menu, select Utilities -> EPIC. Use the QUIT command to exit from your EPIC session. See “QUIT” (page 45).

Accessing EPIC from a command line

You can run a single EPIC command from a UNIX command line or as a cron job. The EPIC command and its parameters are as follows:

```
/opt/nortel/EPIC/bin/epic [-p <epic path>]\
   [<switch> ...] [-e <epic command>]
```

where:

`<epic path>` is the base EPIC configuration directory.

`<switch> ...` is the Passport switch or switches to connect to when you start EPIC. You can add or remove switches by interactive command when EPIC is running. See “EM” (page 39).

`<epic command>` is the EPIC command to be executed. Upon completion of the command, EPIC will exit. The command can be a batch file or a plug-in file.

EPIC session features

During an EPIC session, the switches that are currently connected are displayed in the command line prompt. The configuration parameter, maxprompt, sets a limit to the number of switches that EPIC displays in the prompt. If the number of connected switches exceeds this limit, a summary prompt is displayed that shows the number of connected switches and the status of the connections. See “MAXPROMPT” (page 41).

Every EPIC command is executed on every connected switch unless overridden. See “Overriding connected switches ({})” (page 57).

EPIC manages the connections to the switches. If a connection is lost, a warning message is displayed and EPIC tries to re-establish the connection at ten second intervals. A reconnection message is displayed when EPIC restores the connection.

If multiple connections are established, and sessions have been inactive for over 10 consecutive commands, EPIC automatically closes the inactive sessions. For example, if a user connects to PP1, PP2, and PP3 and issues commands to only PP1, EPIC closes the telnet sessions to PP2 and PP3 if over 10 consecutive commands have been sent to PP1.

During an EPIC session, a cache of the twenty most recently entered commands is maintained. The up and down arrow keys allow you to scroll through this cache so you can select and repeat commands.

EPIC also maintains a cache of the twenty most recently referenced components. Use Ctrl-P keys to scroll through the previous entries and Ctrl-N keys to scroll through the next entries.

Running batch files

Batch files let you automate common tasks. You can run a batch file by using the RUN command or by simply typing its name. You can also include parameters when you run a batch file, such as specifying a batch data file. See "RUN" (page 46) for more information.

You must add the file extension .batch to the name of a batch file. If you omit the .batch extension, you will be unable to run the batch file unless you specify the full path name to the batch file. For example, if you name the batch file addNsta without the .batch extension and store it in directory /home/EPIC/batch, it will not run if you enter the command RUN addNsta unless you specify the full path name RUN EPIC/batch/addNsta

A batch file consists of two types of records, prompts and commands. Prompt records request values to assign to variables. Variable values can be supplied through interactive prompts, from tab-delimited values in a batch data file, or from parameters supplied in the command line.

Command records contain EPIC commands and the variables for which you were prompted. You can include ranges and wildcards in the command records.

Blocks of prompts and commands begin with *;prompts* and *;commands*, respectively.

Batch file example

```
;prompts
Set voice service attributes
============================
Enter range of voice services: <services>
Enter destination Passport: <passport>
Enter destination voice services: <destination>

;commands
add vs/<services>
set vs/<services> framer maxv 32
```

```
set vs/<services> framer minv 24
set vs/<services> framer maxm 64
set vs/<services> framer minm 64
set vs/<services> plc remote "em/<passport>
vs/<destination>"
set vs/<services> plc requiredtx 16000
set vs/<services> plc requiredrx 16000
```

When you run this batch file, you are prompted to supply values for services, passport, and destination. When the commands are executed, the values you supplied are substituted whenever any of these variables occur.

Batch data file

You can supply the values for the prompt variables in a batch data file. This file contains tab-delimited values that EPIC uses to automatically fill in prompt variables. You can generate a batch data file from a spreadsheet application, allowing a whole network to be mapped out as a spreadsheet and provisioned by EPIC.

You can specify a batch data file using the -f<batch data file> option when you execute a batch file. See "RUN" (page 46) for more information.

Batch data file example

```
Local           Switch  Remote
1001-1023       switch1 401-423
1101-1124       switch2 501-523
1201-1223       switch3 501-523
```

> ***Note:*** The first line in this file is skipped because EPIC assumes it is a heading.

If you name this batch data file *vs.txt* and name the file in "Batch file example" (page 29) *sample.batch*, you can use them together by typing the command *sample -fvs.txt* (no space after the -f). EPIC runs *sample* for each line in *vs.txt*, automatically filling in variables with the values in *vs.txt*.

Batch output file

You can direct the commands generated from running a batch file to a batch output file instead of running the commands on Passport switches.

EPIC enhancements in the batch file, such as ranges and wildcards, are expanded into individual CAS commands. In addition, EPIC fills any prompts for variables in the batch file by using interactive prompts, tab-delimited values in a batch data file, or parameters supplied in the command line. These features allow you to run the batch output file outside EPIC.

When you store batch commands to a batch output file, EPIC must be connected to at least one switch. For best results, EPIC is connected to only one switch; otherwise commands are repeated in the batch output file for every connected switch.

You can specify a batch output file using the -o<batch output file> option when you execute a batch file. See “RUN” (page 46) for more information.

Chapter 4
EPIC commands

This section contains detailed information on the Enhanced Passport Interface Controller (EPIC) commands. See the following sections for more information:

About EPIC commands

When you execute a command with parameters in an EPIC session, it overrides any corresponding parameter values that are set in the EPIC configuration file or any default value.

For each command, you only need to enter enough characters for the command to be uniquely identified. These characters are identified by upper-case letters in the syntax definitions, but EPIC itself is not case-sensitive.

EPIC interprets the tab key as a request to complete a partially entered component instance or attribute name. EPIC displays a list of valid completions or if there is only one valid completion, it automatically enters the component instance or attribute name in the command line.

ACTIVATE

ACTivate PRov

This command activates provisioning. It automatically commits, confirms, and saves according to the settings of the AUTOCOMMIT, AUTOCONFIRM, and AUTOSAVE environment variables. It acts the same way as in CAS except that it automatically issues the CONFIRM command after activation is complete in order to avoid accidental rollback of provisioning.

ALARMS

```
ALArms [on|off|hidden]
ALArms [-all] [-s<switch>] [-d<date>] [-t<time>]
[pattern]
```

This command lets you control the alarm collection mode and access buffered alarms. If no options are specified, all alarms received since the last ALARMS command are displayed. The size of the alarm buffer is set using the LOGSIZE configuration file parameter or command.

The command line options are as follows:

`on|off|hidden` specifies the alarm collection mode. The different modes are as follows:

`off` Alarms are not collected.

`on` Alarms are collected and displayed with the switch name added as a prefix. Alarms are also buffered.

`hidden` Alarms are collected and buffered but not displayed. The switch name in the prompt is highlighted to indicate a new alarm has been received.

`-all` displays all alarms stored in the alarm buffer.

`-s<switch>` displays alarms only for the specified switch.

`-d<date>` and `-t<time>` displays all alarms received since the specified date and time. The date format is YYYY-MM-DD, for example, 2000-09-30. The time format is HH:MM:SS, for example, 13:58:26. Both of these formats can be shortened. For example, -d2000-09 displays all alarms received since September 2000.

`pattern` displays all alarms containing the specified pattern.

ALIAS

```
ALIAS [<from>=<to>]
```

This command adds or lists alias definitions. Alias definitions replace the *from* string with the *to* string in EPIC commands before they are executed. If no options are specified, the command lists existing alias definitions. The UNALIAS command removes an alias definition. See “UNALIAS” (page 48).

Example

switch-1> alias no shut=unlock

```
no shut ==> unlock
```

switch-1> alias shut=lock

```
shut ==> lock
```

switch-1> alias

```
no shut ==> unlock
shut ==> lock
```

switch-1> shut sh ca/14

```
Shelf Card/14
ok               2000-02-10 10:00:21.38
```

switch-1>no shut sh ca/14

```
Shelf Card/14
ok              2000-02-10 10:00:28:64
```

AUTOCOMMIT

```
AUTOCOMmit <on|off>
```

When the setting is on, EPIC automatically saves and commits when provisioning is activated.

AUTOCONFIRM

```
AUTOCONfirm <on|off>
```

When the setting is on, EPIC automatically confirms when provisioning is activated.

AUTOSAVE

```
AUTOSave <on|off>
```

When the setting is on, EPIC automatically saves when provisioning is activated.

BEEPER

```
BEEPer <on|off>
```

This command enables or disables beeper warnings. When enabled, the beeper sounds for various conditions, such as when a command that takes more than ten seconds has completed and when switch connections go up or down.

CD

```
CD [-<option>...] [<pathname>]
```

This is a short form for changing directories on a Passport file system. The EPIC command

cd provisioning

is equivalent to the Passport command

cd -p("/provisioning") fs

CHACT

```
CHACT
```

This command checks, activates, and confirms provisioning.

Note: Warning messages from the check are ignored before activating.

CLEAR

```
CLEAR
```

This command clears the screen.

DISPLAY

```
Display [-notab] [-highlight] [-full] <components ...>
```

This command enhances the CAS display command so that results are formatted in tables. This feature simplifies visual comparisons between components.

EPIC identifies each row in the table by a number and each column by a letter. These number and letter combinations are accepted using the # notation to identify attributes in, for example, set commands. See the example in "Reference by table field (#)" (page 56).

The command line options for the DISPLAY command are as follows:

`-notab` turns off table mode for this command. Use this option to display components that already contain formatted tables.

`-highlight` temporarily turns on highlight mode when the default is off. Highlight mode identifies values that have changed since the last DISPLAY command was issued.

`-full` or `-f` causes all attributes to be displayed for the specified components however large the resulting table. Without this parameter, EPIC omits attributes to make the table fit the available window size.

EM

```
EM [<switchname> [<switchname>]...]
EM+ [<switchname> [<switchname>]...]
EM- [<switchname> [<switchname>]...]
{<switchname> [<switchname>]} <command>
```

Use the various forms of this command to control the destination switches for commands. The EPIC prompt shows the currently selected switches in the format *switchname-1 switchname-2 ... switchname-n*.

The different formats of the command are as follows:

`EM` connects to the specified switches, adds them to the prompt, and removes from the prompt any switches that are not specified in the command.

`EM+` connects to the specified switches and adds them to the prompt.

`EM-` removes the specified switches from the prompt.

If a command is prefixed by a switch name or list of switch names that are surrounded by brace brackets ({ }), EPIC sends the command to the specified switch or switches, rather than to the switches listed in the prompt. See “Overriding connected switches ({})” (page 57).

Example

EPIC> em switch-1

```
Connecting...Ready.
```

switch-1> em+ switch-2

```
Connecting...Ready.
```

switch-1 switch-2> em switch-x switch-y switch-z

```
Connecting...Ready.
```

switch-x switch-y switch-z> em- switch-z

switch-x switch-y>

ENV

```
ENV
```

This command causes the current EPIC environment settings, including any defined aliases, to be displayed.

FTP

```
FTP [<switchname> [<switchname>]...]
```

This command allows simultaneous FTP connections to be opened to multiple switches. If no switch names are specified, FTP connects to the list of switches shown in the EPIC prompt.

This command is intended for the application of software patches or for downloading provisioning files. Once an FTP connection is opened, the usual FTP commands such as pwd, cd, get, and put are accepted. EPIC implements login information and macros defined in the .netrc file in your home directory.

To exit FTP mode, enter the EM command without specifying any switches.

HELP

Help <topic>

This command displays user information on the specified topic. If EPIC does not recognize the specified topic, it activates CAS help instead.

HIGHLIGHT

```
HIGHlight <on|off>
```

This command sets the highlight mode. When on, values that have been changed between DISPLAY commands are highlighted.

LOG

```
LOG <filename>
```

This command starts logging the EPIC session to the specified file. Make sure you have UNIX permissions to write files to the target directory. See also “UNLOG” (page 48).

LOGSIZE

```
LOGSIZE [<number of alarms>]
```

This command sets the number of alarms that are retained in the alarm buffer. When the number of alarms received exceeds the buffer size, the oldest alarms are discarded.

LS

```
LS [-<option>...] [<pathname>]
```

This is a short form for listing directories on a Passport file system. The EPIC command

ls provisioning

is equivalent to the Passport command

ls -p(“/provisioning”) fs

When connected to Passports using the FTP command, this short form does not apply.

MAXPROMPT

```
MAXPROMPT [<number of switches>]
```

This command sets a limit to the number of switch names that can be displayed in the EPIC command line prompt. Once more than the set number of switches are connected, EPIC summarizes the prompt by displaying the number of switches connected and the switch modes, if they apply.

To obtain detailed information about connected switches, see “STATUS” (page 47).

Examples

switch1 switch2 switch3 switch4> maxprompt 3

```
The prompt will now show up to 3 switches.
```

4-Switches

Examples of displayed switch modes:
7-Switches (OS) >
4-Switches (3-PROV)>
15-Switches (1-DOWN) (PROV)>

MULTISWITCH

MULTISWITCH [<on|off>]

This command sets multiswitch mode. By default, EPIC allows connection with large numbers of switches at the same time and sends commands to them simultaneously. If multiswitch mode is off, EPIC allows connection to only one switch at a time.

ONALL

```
ONall <onall file> <command>
```

This command allows you to run commands on all the switches listed in the onall file, which you create. See “Onall file” (page 24).

When an ONALL command is issued, EPIC connects to as many switches listed in the onall file as possible and runs the specified command on all of them at once. The command can be a batch file or a plug-in.

Example

EPIC> more testlab.onall

```
switch-1
switch-2
```

EPIC> onall testlab d ns

```
Connecting...Ready.
==|=========================================
 #|                  SWITCH-1       SWITCH-2
```

```
  |                 NS            NS
==|=================(A)===========(B)==========
 1|adminStat        unlocked      unlocked
 2|operationalState enabled       enabled
 3|usageState       busy          busy
 4|clockSyncState   synchronized  synchronized
 5|activeReference  Lp/10 Sonet/2 Lp/4 DS1/0
 6|standbyReference               Lp/11 DS1/0
==|============================================
```

PAGER

```
PAGEr <on|off>
```

This command controls pagination. When it is on and any EPIC command response fills more than one screen, the output is displayed one screen at a time and you are prompted to press any key to display the next screen.

PING

```
PING [vr/<virtual router number>] <ip address>
```

This command is a simplified version of the standard CAS PING command. If you do not specify a virtual router number, EPIC assumes vr/1.

PLUGINSEQ

```
PLUGINSEQ [<on|off>]
```

This command determines whether plug-ins run sequentially on the switches. When on, plug-ins run on the selected Passport switches one at a time. When off, plug-ins are run simultaneously on the selected Passport switches.

POLL

```
POLL [-f<filename>] [-<delay>] [-change] [-value]\
     [-round] [-%<percent of>] [-bg]\
     [-stop <process-id>] <attributes>
```

This command provides a method for monitoring Passport components. If no options are specified, EPIC lists background POLL processes.

The advantage of POLL over the WATCH command is that POLL can monitor a very large number of components at the same time, more than would comfortably fit on a WATCH screen. In addition, POLL stores its results for later reference. Results are stored in tab-delimited format to simplify importing them into a spreadsheet application.

Normally POLL continues polling until you press any key. If you use the -bg option, POLL continues until you enter the -stop option.

The command line options are as follows:

`-f<filename>` is the file in which POLL results are stored. If you do not specify this option, the POLL command prompts you for a filename. The current date is appended to the filename. When a date change occurs, the current file is closed and a new file is opened.

`-<delay>` specifies the time, in seconds, between polls. If you do not specify this option, a delay of 10 seconds is assumed.

> ***Note:*** If the delay option is set too low, the POLL command may adversely affect the performance of the control processor of the Passport switch on which the command is being executed.

`-change` causes only the change between polls to be recorded; components are not recorded unless there has been a change.

`-value` records raw values rather than rates-per-second. By default, values reported by POLL are in rates-per-second. So if you do not specify this option, for each poll period, POLL displays the selected components, subtracts the previous value, and divides by the time difference between polls.

`-round` causes rates-per-second to be rounded to the nearest integer.

`-%<percent of>` lets you record results as a percentage of a set value. For example,

```
poll -%96000 atmif/10 txcell
```

polls atmif/10 on all your switches and stores the results as percentages of 96000 (a full DS3).

`-bg` runs POLL in the background. EPIC reports the process-id that is running the POLL and then accepts new commands. You can list POLL commands running in the background by entering POLL with no options.

`-stop <process-id>` stops a POLL process that is running in the background.

Example

switch-1 switch-2> poll -fatmiftest -bg atmif/* stats

```
Values will be recorded as rates per second.
Polling in the background with process-ID 10208.
Re-Connecting...Ready.
```

switch-1 switch-2> poll

```
10208 Log-file:   atmiftest
      Interval:   10 seconds
      Mode:       Rate/s
      Switches:   switch-1 switch-2
      Attributes: atmif/* stats

Type "help poll" for more information.
```

switch-1 switch-2> poll -stop 10208

```
POLL process 10208 stopped.
```

QUIT

QUIT|BYE|EXIT|LOGOFF|LOGOUT

This command causes EPIC to log out from all connected switches and then exit. EPIC accepts the above alternatives to QUIT.

ROUTE

```
ROute [vr/<virtual router number>] [<ip address>]
```

This command searches the ILS routing database for specified addresses, based on both IP address and subnet mask. If you do not specify a virtual router number, EPIC assumes vr/1. If you do not specify an IP address, EPIC displays the whole routing table.

RUN

```
RUN
RUN [-f<batch data file>] [-o<batch output file>]\
    <batch file> [<argument>...]
```

This command runs a batch file. You can also run a batch file simply by typing its name. If no options are specified, EPIC lists available batch files.

The command line options are as follows:

`-f<batch data file>` specifies the name of the batch data file. When you use this option, EPIC fills prompts in the batch file with values from this file. See “Batch data file” (page 30) for more information.

`-o<batch output file>` specifies the name of the batch output file. When you use this option, EPIC stores the batch commands in this file rather than running the commands on Passport switches. See “Batch output file” (page 30) for more information.

`batch file` is the name of the batch file to be executed. You must include a file extension of .batch to a batch file. See “Batch files” (page 21) and “Running batch files” (page 29) for more information.

`<argument>...` is one way you can supply values to prompts in the batch file. You can also supply values by using a batch data file or by responding to interactive prompts.

SCOM

```
SCOM
```

This is a short form to save and then commit provisioning.

SHELL

```
SHELL [<UNIX command>]
```

This command opens up a UNIX shell and runs the specified UNIX command. If you do not specify a command, an interactive shell is opened.

STATUS

```
STATus
```

This command displays a list of connected switches and the current mode of each switch. Switch modes include Provisioning, Operational, FTP, OS, and DOWN. This command also displays the maximum number of simultaneous connections and the current connection totals.

Example

switch-1 switch-2 switch-3 switch-4> stat

```
CONNECTION MODE
---------- ----
  switch-1 Provisioning
  switch-2 Operational
  switch-3 Provisioning
  switch-4 Provisioning

Maximum connections: 1017
Active connections: 4
Idle connections: 0
```

TABLES

```
TABLES <on|off>
```

This command sets the table mode.

> ***Note:*** You can also turn off table mode on a command-by-command basis when you use the -notab option of the DISPLAY command.

TELNET

TELNET [vr/<virtual router number>] <ip address>

This command opens a Telnet connection from an ILS virtual router to a specified address. EPIC supports only one Telnet connection at a time so before you issue this command, you may have to issue the EM command to select one switch. If you do not specify a virtual router number, EPIC assumes vr/1.

UNALIAS

```
UNALIAS <from>
```

This command removes the specified <from> alias definition.

Example

EPIC> alias

```
shut ==> lock
```

EPIC> unalias shut

```
Alias 'shut' removed.
```

EPIC> alias

```
The alias buffer is empty.
```

UNLOG

```
UNLOG
```

This command stops logging to a file after a log has been opened with the LOG command.

USER

```
USER <userid> [<password>]
```

This command specifies the userid and password to use to connect to all Passport switches. If any connections are open when you issue this command, EPIC logs out from all switches and logs back in using the specified userid and password. The userid and password are also used when establishing new connections.

If you do not specify a password, EPIC prompts you to enter one.

WATCH

WATCH [-<delay in seconds>] <components...>

This command repeatedly displays the specified components and their attributes. EPIC refreshes the display regularly, either at the default rate of 2 seconds or else using the rate in the -<delay in seconds> option.

Note: If the -<delay in seconds> option is set too low, the WATCH command may adversely affect the performance of the control processor of the Passport switch on which the command is being executed.

WATCH runs for all of the switches in the EPIC prompt, which allows simultaneous monitoring of components on multiple Passport switches.

Command output resembles output from EPIC's DISPLAY command except that a variety of statistics are displayed instead of just the attribute values. A menu of statistics is displayed at the top of the screen, letting you select the statistics to include in the displayed table. Available statistics include current, minimum, and maximum rates per second; minimum and maximum value; total changes since WATCH started; average value; average rate per second; and current value.

The following keys initiate the indicated actions:

`C` clears all statistics.

`M` specifies that the display is to be refreshed only when you press the space bar.

`H` toggles highlighting of changed statistics on and off.

Up and down arrow keys scroll the WATCH display if the window is too small to show all selected components.

`Q` exits WATCH mode and returns to EPIC command line processing.

Example

```
switch-1 switch-2> watch atmif/20-21
```

```
1=Rate/s 2=Min/s 3=Max/s 4=Min 5=Max 6=Change 7=Average 8=Average/s
                     9=Current C=Clear M=Manual H=Highlight Q=Quit
==|=============================================================
 #|                              SWITCH-1 SWITCH-1 SWITCH-2 SWITCH-2
  |                              AtmIf/20 AtmIf/21 AtmIf/20 AtmIf/21
==|============================(A)======(B)======(E)======(F)=====
 1|adminState                   unlocked unlocked unlocked unlocked
```

```
 2|operationalState              enabled  enabled  enabled  enabled
 3|usageState                    active   active   active   active
 4|availabilityStatus
 5|proceduralStatus
 6|controlStatus
 7|alarmStatus
 8|standbyStatus                 notSet   notSet   notSet   notSet
 9|unknownStatus                 false    false    false    false
10|unshapedTransmitQueueingOper common   common   common   common
11|transmittedCells              8.73     10.38    9.54     6.11
12|txLinkUtilization             0.00     0.00     0.00     0.00
13|receivedCells                 12.89    6.23     7.88     7.74
14|rxLinkUtilization             0.00     0.00     0.00     0.00
15|droppedRxCells                0.00     0.00     0.00     0.00
16|lastDroppedRxCellConnection   0.00     0.00     0.00     0.00
17|aal5RxErrors                  0.00     0.00     0.00     0.00
18|lastAal5RxErrorConnection     0.00     0.00     0.00     0.00
19|lrcFrameErrors                0.00     0.00     0.00     0.00
20|lastLrcFrameErrorConnection   0.00     0.00     0.00     0.00
==|============================================================
```

WIDTH

```
WIDTH [<screen width>]
WIDTH [AUTO]
```

This command displays or changes the EPIC terminal width. If no options are specified, the current width is displayed.

EPIC detects the width automatically, but this command lets you override the default width, for example, when logging results to a file. The minimum terminal width is 80 characters.

The WIDTH AUTO command resumes automatic width detection.

WILDCARD

```
WILDCARD [smart|all]
```

EPIC's special wildcards (**) are more flexible than standard CAS wildcards (*). However special wildcards can affect performance and should not be used unless they are necessary. By selecting smart as the WILDCARD option, EPIC automatically ignores special wildcards for LS and DISPLAY commands. See also "Special wildcards (**)" (page 55).

Chapter 5
EPIC special characters

This section contains information on EPIC functions that require special characters in the command. See the following sections for more information:

Numeric ranges (-)

Use the hyphen (-) to specify ranges of numeric components in your commands. The command that contains the range is executed on every component in the range.

Example

switch-1> d vs/501-503 framer maxvoice, minvoice

```
==|=======================================
 #|                        maxVoice  minVoice
  |                        G711G726  G711G726
  |                        Rate      Rate
==|======================(1)=======(2)======
 A|SWITCH-1 Vs/501 Framer 32 kbit/s 24 kbit/s
```

```
 B|SWITCH-1 Vs/502 Framer 32 kbit/s 24 kbit/s
 C|SWITCH-1 Vs/503 Framer 32 kbit/s 24 kbit/s
==|=======================================
```

If you specify two ranges in the same command, both ranges must span the same number of values and EPIC assumes that they correlate with each other. For example

`set vs/501-505 plc remote "em/switch-1 vs/521-525"`

correlates vs/501 with vs/521, vs/502 with vs/522, etc.

You can also use ranges to specify ranges of consecutive VCI values.

Example

switch-1> add atmif/70 vcc34.32-40

```
AtmIf/70 Vcc/34.32
   The following components have been created:
      AtmIf/70 Vcc/34.32 Vcd
      AtmIf/70 Vcc/34.32
AtmIf/70 Vcc/34.33
   The following components have been created:
      AtmIf/70 Vcc/34.33 Vcd
      AtmIf/70 Vcc/34.33
AtmIf/70 Vcc/34.34
   The following components have been created:
      AtmIf/70 Vcc/34.34 Vcd
      AtmIf/70 Vcc/34.34
etc...
```

EPIC ignores the range notation in IP addresses since the hyphen in an IP address can sometimes indicate a port number. To specify a range in an IP address, use two hyphens (--). For example

`add vr/carr01 pp/vm0 ipp lo/200.1.2--150.1`

Non-sequential components (+)

Use the plus sign (+) to specify multiple, non-contiguous components.

Example

switch-1> set vs/501-503+510-513 framer silencesup on

```
Vs/501 Framer
Vs/502 Framer
Vs/503 Framer
Vs/510 Framer
Vs/511 Framer
Vs/512 Framer
Vs/513 Framer
```

Instance selection (&)

Use the and sign (&) when components have different instance numbers on each switch but you want to issue commands against them at the same time. Specify the component numbers in the same order as their switches appear in the EPIC prompt.

Example

switch-1 switch-2> d atmif/100&70 conmap

```
==|=========================================================
 #|                        SWITCH-1          SWITCH-2
  |                        AtmIf/100 ConnMap AtmIf/70 Connmap
==|=======================(A)===============(B)==============
 1|numVccsForVpiZero       256               256
 2|numNonZeroVpisForVccs   4                 4
 3|firstNonZeroVpiForVccs  32                32
 4|numVccsPerNonZeroVpi    256               256
==|=========================================================
```

Special wildcards (**)

CAS allows wildcards (*) on multiple hierarchy levels for the LIST and DISPLAY commands but not for other commands such as SET and LOCK. You can overcome this limitation by using EPIC wildcards (**) on any command that refers to multiple levels in the CAS hierarchy.

Example

switch-1> set atmif/ trafficshaping enabled**

```
AtmIf/70
ok
```

```
AtmIf/71
ok
AtmIf/100
ok
AtmIf/101
ok
```

Reference by table field (#)

Use the octothorpe (#) to reference entries in the last displayed table. Columns are represented by letters and rows by numbers, for example #a2.

You can reference a specific attribute on every displayed component by using just the attribute number, for example, #2.

Example

switch-1 switch-2> d atmif/70 connmap

```
==|=========================================================
 #|                        SWITCH-1          SWITCH-2
  |                        AtmIf/70 ConnMap  AtmIf/70 Connmap
==|=======================(A)==============(B)==============
 1|numVccsForVpiZero       256               256
 2|numNonZeroVpisForVccs   4                 4
 3|firstNonZeroVpiForVccs  32                32
 4|numVccsPerNonZeroVpi    256               256
==|=========================================================
```

switch-1 switch-2> set #1 768 #2 1

```
SWITCH-1 AtmIf/70 ConnMap
ok
SWITCH-2 AtmIf/70 ConnMap
ok
```

switch-1 switch-2> set #a3 48

```
SWITCH-1 AtmIf/70 ConnMap
ok
```

Overriding connected switches ({})

Every EPIC command is executed on every connected switch unless overridden. You can run a command on another switch or set of switches by prefixing the command with a list of switches in brace brackets ({}). After the command has completed, you are returned to your original set of switches.

Example

switch-x switch-y> {switch-1} l fruni/*

```
Connecting...Ready.
FrUni/80
FrUni/82
ok
```

Disabling EPIC enhancements (~)

If an EPIC command is preceded by the tilde (~), EPIC command line interpretation is not applied to the command. Use this feature if EPIC notation, such as the hyphen (-), and sign (&), or plus sign (+) are interfering with the command you are running.

Example

switch-1> ~set trk/10 expectedremote “a & b”

```
Trk/10
ok
```

Index

Preside Multiservice Data Manager

EPIC

Reference Guide

Release R15.1

Publication: 241-6001-809
Document status: Standard
Document version: 15.1RSUP
Document date: August 2004
Printed in Canada

NORTEL NETWORKS