Log System Management Commands

This chapter describes the command-line interface (CLI) commands that you can use to debug the Prime Cable Provisioning Device Provisioning Engine (DPE), and monitor and manage the Prime Cable Provisioning log system.

Before using a debug command, you must enable DPE debugging by running the **debug on** command. If you run the following commands on an unlicensed DPE, a message similar to this one appears:

```
This DPE is not licensed. Your request cannot be serviced. Please check with your
system administrator for a DPE license.
```

Caution Enabling debug logging may have a severe impact on DPE performance. Do not leave the DPE running with debug turned on for an extended period of time.

The commands described in this chapter are:

Command	Description	CLI Mode		Required Privileges			
		Login	Enable	PRIV_DPE_READ	PRIV_DPE_UPDATE	PRIV_DPE_SECURITY	PRIV_DEVICE_READ
clear logs	Removes out-of-date log files from the system.		✓	✓	✓		
debug dpe cache	Debugs the DPE cache.		✓	✓	✓		
debug dpe connection	Debugs the DPE connection.		✓	✓	✓		
debug dpe dpe-server	Debugs the DPE server.		✓	✓	✓		
debug dpe event-manager	Debugs the DPE event manager.		✓	✓	✓		
debug dpe exceptions	Debugs DPE exceptions.		✓	✓	✓		
debug dpe framework	Debugs the DPE framework.		✓	✓	✓		

Command	Description	CLI Mode		Required Privileges			
		Login	Enable	PRIV_DPE_READ	PRIV_DPE_UPDATE	PRIV_DPE_SECURITY	PRIV_DEVICE_READ
debug dpe messaging	Debugs DPE messaging.		✓	✓	✓		
debug dpe ssl_all	Enables the JSSE internal messaging category for debugging ssl messages		✓	✓	✓		
debug dpe secure_messaging	Enables the ssl messaging category for debug messages		✓	✓	✓		
debug on	Enables debug logging.		✓	✓	✓		
debug service tftp ipv4 | ipv6	Debugs TFTP transfers.		✓	✓	✓		
no debug all	Disables debug logging.		✓	✓	✓		
log level	Sets the level of minimum DPE log messages.		✓	✓	✓		
show log	Displays recent log entries for the DPE.	✓	✓	✓			

clear logs

Use the **clear logs** command to remove historic (out-of-date) log files that exist on the system. These files include:

- DPE logs
- Hardware
- Syslog

Over time, historic log files accumulate within the DPE. You can use the **support bundle state** command to bundle these logs. We recommend that you create a bundle before clearing logs, so that no necessary files are lost accidently.

Syntax Description

No keywords or arguments.

Defaults No default behavior or values.

Examples

```
bac_dpe# clear logs
Clearing historic log files...
+ Removing 1 DPE log files...
+ No more historic logs.
```

debug dpe

Use the **debug dpe** command to configure debug settings on the DPE. Table 6-1 describes the keywords that you can use with this command.

Note Enter the commands described in Table 6-1 as indicated.

Table 6-1 List of debug dpe Commands

Command	Description	
debug dpe cache **no debug dpe cache**	Enables debugging of the DPE cache, which involves messages pertaining to the DPE cache including: • Logging requests for cache entries • Updates to the cache • Other interactions by DPE subsystems To disable DPE cache debugging, use the **no** form of this command.	

	Examples
	This result occurs when you enable debugging of the DPE cache.
	`bac_dpe# debug dpe cache`
	`% OK`
	
	This result occurs when you disable debugging of the DPE cache.
	`bac_dpe# no debug dpe cache`
	`% OK`
		Defaults
		Debugging of the DPE cache is by default disabled.

Table 6-1 List of debug dpe Commands (continued)

Command	Description	
debug dpe connection **no debug dpe connection**	Enables the debugging of the DPE connection, which logs communication subsystem status and error messages. Use this command to identify communication problems between the DPE and the RDU. To disable debugging of the DPE connection, use the **no** form of this command.	

	Examples
	This result occurs when you enable debugging of the DPE connection.
	`bac_dpe# debug dpe connection`
	`% OK`
	
	This result occurs when you disable debugging of the DPE connection.
	`bac_dpe# no debug dpe connection`
	`% OK`
		Defaults
		Debugging of the DPE connection is by default disabled.

debug dpe dpe-server **no debug dpe dpe-server**	Enables debugging of the DPE server, which involves logging messages about the overall status and issues of the DPE server. To disable the debugging of the DPE server, use the **no** form of this command.	

	Examples
	This result occurs when you enable debugging of the DPE server.
	`bac_dpe# debug dpe dpe-server`
	`% OK`
	
	This result occurs when you disable debugging of the DPE server.
	`bac_dpe# no debug dpe dpe-server`
	`% OK`
		Defaults
		Debugging of the DPE server is by default disabled.

debug dpe event-manager **no debug dpe event-manager**	Enables debugging of the DPE event manager, which involves logging messages and conditions showing the state of the event manager. To disable debugging of the DPE event manager, use the **no** form of this command.	

	Examples
	This result occurs when you enable debugging of the DPE event manager.
	`bac_dpe# debug dpe event-manager`
	`% OK`
	
	This result occurs when you disable debugging of the DPE event manager.
	`bac_dpe# no debug dpe event-manager`
	`% OK`
		Defaults
		Debugging of the DPE event manager is by default enabled.

Table 6-1 List of debug dpe Commands (continued)

Command	Description	
debug dpe exceptions **no debug dpe exceptions**	Enables the debugging of DPE exceptions, which involves logging full stack traces for exceptions occurring during system operation. In unusual situations, such as when the system is apparently corrupt or behaving abnormally, this command can provide valuable information for Cisco support. To disable the debugging of DPE exceptions, use the **no** form of this command.	

	Examples
	
	This result occurs when you enable debugging of DPE exceptions.
	
	`bac_dpe# debug dpe exceptions`
	`% OK`
	
	This result occurs when you disable debugging of DPE exceptions.
	
	`bac_dpe# no debug dpe exceptions`
	`% OK`
		Defaults
		
		Debugging of DPE exceptions is by default enabled.

debug dpe framework **no debug dpe framework**	Enables the debugging of the DPE framework, which involves logging information about the underlying framework of the DPE server. This infrastructure provides for all the various servers in Prime Cable Provisioning. To disable the debugging of the DPE framework, use the **no** form of this command.	

	Examples
	
	This result occurs when you enable debugging of the DPE framework.
	
	`bac_dpe# debug dpe framework`
	`% OK`
	
	This result occurs when you disable debugging of the DPE framework.
	
	`bac_dpe# no debug dpe framework`
	`% OK`
		Defaults
		
		Debugging of the DPE framework is by default enabled.

Table 6-1 List of debug dpe Commands (continued)

Command	Description	
debug dpe messaging **no debug dpe messaging**	Enables debugging of DPE messaging, which involves logging details about the DPE messaging subsystem. This subsystem is used primarily for communication between the DPE and the RDU. To disable the debugging of DPE messaging, use the **no** form of this command.	

	Examples
	
	This result occurs when you enable debugging of DPE messaging.
	
	`bac_dpe# debug dpe messaging`
	`% OK`
	
	This result occurs when you disable debugging of DPE messaging.
	
	`bac_dpe# no debug dpe messaging`
	`% OK`
		Defaults
		
		Debugging of DPE messaging is by default disabled.

debug dpe ssl_all **no debug dpe ssl_all**	Enables the detailed JSSE internal messaging category for debugging SSL messages To disable the internal debugging of JSSE, use the no form of this command.	

	Examples
	
	This result occurs when you enable debugging of JSSE internal messaging category.
	
	`bac_dpe# debug dpe ssl_all`
	`% OK`
	
	This result occurs when you disable debugging of JSSE internal messaging category.
	
	`bac_dpe# no debug dpe ssl_all`
	`% OK`
		Defaults
		
		Debugging of JSSE internal messaging is by default disabled.

debug dpe secure_messaging **no debug dpe secure_messaging**	Enables the basic SSL messaging category for debug messages. To disable SSL messaging category for debug messages.	

	Examples
	
	This result occurs when you enable debugging of basic SSL connections.
	
	`bac_dpe# debug dpe secure_messaging`
	`% OK`
	
	This result occurs when you disable debugging of basic SSL connections.
	
	`bac_dpe# no debug dpe secure_messaging`
	`% OK`
		Defaults
		
		Debugging of basic SSL connections is by default disabled.

debug on

Use the **debug on** command to enable debug logging, which can be helpful when troubleshooting possible system problems. Additionally, you must separately enable specific debugging categories with commands such as **debug dpe cache**.

Caution Enabling debug logging may have a severe impact on DPE performance. Do not leave the DPE running with debug turned on for an extended period of time.

To disable all the categories of debug logging, run the **no debug all** command. See no debug all, page 6-9.

Syntax Description

No keywords or arguments.

Defaults

Debugging is by default disabled.

Examples

```
bac_dpe# debug on
% OK
```

debug service tftp ipv4 | ipv6

Use the **debug service tftp ipv4 | ipv6** command to enable debugging of TFTP transfers for IPv4 or IPv6.

To disable debugging of the TFTP service, use the **no** form of this command. See no debug service tftp ipv4 | ipv6, page 6-8.

Syntax Description

debug service tftp *1* **ipv4 | ipv6**

- *1*—Identifies the instance of the TFTP service on the DPE.
- **ipv4**—Specifies debugging of the TFTP service for IPv4.
- **ipv6**—Specifies debugging of the TFTP service for IPv6.

Defaults

Debugging of the TFTP service is by default disabled.

Examples

This result occurs when you enable debugging of the TFTP service for IPv4.

```
bac_dpe# debug service tftp 1 ipv4
% OK
```

This result occurs when you enable debugging of the TFTP service for IPv6.

```
bac_dpe# debug service tftp 1 ipv6
```

```
% OK
```

no debug service tftp ipv4 | ipv6

Use the **no debug service tftp ipv4 | ipv6** command to disable debugging of TFTP transfers for IPv4 or IPv6.

To enable debugging of the TFTP service, see debug service tftp ipv4 | ipv6, page 6-7.

Syntax Description

no debug service tftp *1* **ipv4 | ipv6**

- *1*—Identifies the instance of the TFTP service on the DPE.
- **ipv4**—Specifies debugging of the TFTP service for IPv4.
- **ipv6**—Specifies debugging of the TFTP service for IPv6.

Defaults

Debugging of the TFTP service is by default disabled.

Examples This result occurs when you disable debugging of the TFTP service for IPv4.

```
bac_dpe# no debug service tftp 1 ipv4
% OK
```

This result occurs when you disable debugging of the TFTP service for IPv6.

```
bac_dpe# no debug service tftp 1 ipv6
% OK
```

no debug all

Use the **no debug all** command to disable all the categories of debug logging.

For details about enabling debug logging, see debug on, page 6-7.

Syntax Description No keywords or arguments.

Defaults Debug logging is by default disabled.

Examples

```
bac_dpe# no debug all
% OK
```

log level

Use the **log level** command to set the level of minimum DPE log messages that are saved, as described in the *Cisco Prime Cable Provisioning 7.2.1 User Guide*.

Syntax Description **log level** *number*

number—Identifies the logging level, by number, to be saved. Table 6-2 describes the log levels that Prime Cable Provisioning supports.

Table 6-2 DPE Log Levels

Log Level No.	Description
0-emergency	Saves all emergency messages.
1-alert	Saves all activities that need immediate action and those of a more severe nature.
2-critical	Saves all critical conditions and those of a more severe nature.
3-error	Saves all error messages and those of a more severe nature.
4-warning	Saves all warning messages and those of a more severe nature.

Table 6-2 DPE Log Levels (continued)

Log Level No.	Description
5-notification	Saves all notification messages and those of a more severe nature.
6-info	Saves all logging messages available.

Note Setting a specific log level saves messages less than or equal to the configured level. For example, when you set the log level at 5-notification, all events generating messages with a log level of 4 or less are written into the log file.

The logging system's log levels are used to identify the urgency with which you might want to address log issues. The 0-emergency setting is the most severe level of logging, while 6-info is the least severe, saving mostly informational log messages.

Defaults The default log level is 5-notification.

Examples

```
bac_dpe# log level 6
% OK
```

show log

Use the **show log** command to show all recent log entries for the DPE. These logs contain general DPE process information, including all system errors or severe problems. Check this log when the system is experiencing difficulties.

If the log contains insufficient information, enable the debug logging function and experiment with the different categories related to the problem. See debug dpe, page 6-3, for detailed information.

Syntax Description **show log** [**last** *1..999* | *run*]

- **last** *1..999*—Shows the specified number of recent log entries for the DPE, with *1..999* specifying the number of log entries that you want to display. This keyword is optional.
- **run**—Displays the running DPE log, which starts showing all messages logged to the DPE log. The command continues to run until you press Enter. This keyword is optional.

Defaults No default behavior or values.

Examples This result occurs when you use the **show log** command.

```
bac_dpe# show log
dpe.example.com: 2007 06 04 08:01:42 EDT: %BPR-DPE-5-0236: [Device Provisioning Engine]
starting up.
```

```
dpe.example.com: 2007 06 04 08:01:42 EDT: %BPR-DPE-6-0822: Server version [Cisco Prime
Cable Provisioning 5.1 (SOL_BAC5_1_0_00000000_0505)].
dpe.example.com: 2007 06 04 08:01:42 EDT: %BPR-DPE-6-0689: Maximum Java heap size [307
MiB].
dpe.example.com: 2007 06 04 08:01:42 EDT: %BPR-DPE-6-0690: Maximum database cache size
[102 MiB].
dpe.example.com: 2007 06 04 08:01:42 EDT: %BPR-DPE-5-1360: Connecting to RDU
[dpe.example.com:49187]. Rate [1/d].
dpe.example.com: 2007 06 04 08:05:31 EDT: %BPR-DPE-5-0195: Connected to RDU
[dpe.example.com:49187]. Time to connect [3.8 min]. Rate [1/d].
dpe.example.com: 2007 06 04 08:05:31 EDT: %BPR-DPE-5-0982: Configured provisioning
interfaces: [localhost[10.10.0.1]].
dpe.example.com: 2007 06 04 08:05:31 EDT: %BPR-DPE-5-1359: Batch
[DPE:dpe.example.com/10.86.149.133:bf7190:112f6a01cf7:80000002]. Registering with RDU.
Rate [1/d].
dpe.example.com: 2007 06 04 08:05:32 EDT: %BPR-LICENSING-3-0998: Server registration
failed. Lack of DPE licenses.
dpe.example.com: 2007 06 04 08:05:33 EDT: %BPR-DPE-5-1374: Opening database [default.db].
dpe.example.com: 2007 06 04 08:05:34 EDT: %BPR-DPE-5-1375: Opened database [default.db].
Time to open [1.2 s].
dpe.example.com: 2007 06 04 08:05:34 EDT: %BPR-TFTP-5-0462: Service is disabled.
dpe.example.com: 2007 06 04 08:05:34 EDT: %BPR-TOD-5-5501: TOD Server disabled.
dpe.example.com: 2007 06 04 08:19:21 EDT: %BPR-LICENSING-5-1002: DPE received a license
event from the RDU.
dpe.example.com: 2006 12 21 11:22:20 GMT: %BPR-DPE-5: DPE-0: Device Provisioning Engine
starting up
...
```

Note The output presented in this example is trimmed for demonstration purposes.

This result occurs when you use the **show log last** command.

```
bac_dpe# show log last 2
dpe.example.com: 2007 06 04 08:19:23 EDT: %BPR-DPE-5-0147: Batch dpe.example.com: 2007 06
04 08:19:23 EDT: %BPR-DPE-5-1371: Synchronized [0] cached device configurations with RDU.
Time to synchronize [52 ms] ([0/s]).
dpe.example.com: 2006 12 21 11:28:17 GMT: %BPR-DPE-5: DPE-0: Device Provisioning Engine
starting up
```

This result occurs when you use the **show log run** command.

```
dpe# show log run
Press <enter> to stop.
dpe.example.com: 2006 12 21 11:43:43 GMT: %BPR-DPE-5: DPE-0: Device Provisioning Engine
starting up
dpe.example.com: 2006 12 21 11:43:44 GMT: %BPR-DPE-5: Info DPE: Attempt to connect to RDU
BPR_host.example.com:49187 failed;
dpe.example.com: 2006 12 21 11:43:44 GMT: %BPR-DPE-5: Info TFTP: Ready to service requests

% Stopped.
```