Basic Troubleshooting Techniques

SmartEdge OS Software

FAULT TRACING DIRECT.

Copyright

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List

SmartEdge	is a registered trademark of Telefonaktiebolaget LM Ericsson.

Contents

1 Overview 1

2 Collecting Troubleshooting Data 3

2.1 Collecting the Output of Logs and Show Commands 3

2.2 Enabling Terminal Emulator Logging 4

3 Getting Help for a Command 9

4 Contexts in Debugging 11

4.1 How the Active Context Affects Debug Output 11

4.2 Identifying Context-Specific Debug Functions 13

4.3 Displaying Debug Output through the Craft Port 14

4.4 Displaying Debug Output Through Telnet or SSH 14

4.5 Debugging Tasks 15

5 Using Basic Troubleshooting Commands 17

6 Accessing the SmartEdge System Components 19

6.1 Access Primary and Secondary XCRP Controller Cards 19

6.2 Logging On to the Standby XCRP Controller Card 20

6.3 Accessing NetBSD Shell Mode 21

6.4 Accessing and Using Open Firmware (OFW or OpenBoot) Mode 21

7 Backing Up Configurations and Data in Memory 31

1 Overview

This document describes how to get help with a command, the role of contexts in troubleshooting, how to perform basic debugging tasks, and use basic troubleshooting commands. It also describes how to access the SmartEdge® router components, perform backups, collect troubleshooting data, and enable logging.

The audience for this document is the general SmartEdge OS user.

For information about troubleshooting specific problems, see the following documents:

- *ASE Troubleshooting Guide*
- *BRAS Troubleshooting Guide*
- *Debugging*
- *General Troubleshooting Guide*
- *Troubleshooting MPLS*
- *Troubleshooting OSPF*
- *Troubleshooting IPv6 and Dual-Stack Subscriber Services*
- *Troubleshooting IS-IS*
- *Troubleshooting L3VPNs*
- *Troubleshooting VPLS*

2 Collecting Troubleshooting Data

Before you begin troubleshooting, gather the evidence of what has been happening on your router. Collect the output of the **`show tech-support`** command (and optionally other show commands and macros for specific problems). This evidence should be collected before beginning to troubleshoot because some troubleshooting techniques destroy or modify already stored data. If you need to escalate your problem to customer support, you need to include troubleshooting data with your support request. For guidelines on how to collect the required data, see *Data Collection Guideline for the SmartEdge Router*.

2.1 Collecting the Output of Logs and Show Commands

There are several ways to collect the output of show commands, logs, and macros:

- To save the output of show commands to `/flash` or `/md` before copying it to a remote location, add **`| save file`** ***`filename`*** or **`| save file /md/`*****`filename`*** keywords to the end of the command. For example to save the output of the **`show redundancy`** command in `/flash/show-redun.txt`, use the , **`show redundancy | save file show-redun.txt`** command.

- to save your CLI session to a file on your computer, use the capture or logging function in your terminal emulation software; for an example of enabling logging on a terminal emulation application, see Section 2.1 on page 3.

- Use the UNIX **`script`** command on the terminal server (before logging onto the router and running the show command) to save the output to a file in your working directory.

For example:

To save the output of the **`show tech-support`** command to `/md` and then to an external drive:

1. Enter the **`show tech-support | save /md/filename`** command.

   For example, to save the output to the `showtech.txt` file on an external CF card, enter the following command:

   ```
   show tech-support | save /md/showtech.txt
   ```

2. To copy the output file to a remote location, use the **`copy /md/showtech.txt ftp://`*****`username@hostname`*****`/showtech.txt`** command.

To use the **script** command to save the output to a file in your working directory:

1. Accessing the router from a UNIX environment (for example, from a terminal server), enter the **script** ***filename*** command.

2. Telnet to the router and log on.

3. Enter the **show tech-support** command.

   Your session is saved to a file in your working directory. For example, to save the output of the command on the router, `isp-224` (with the IP address `10.10.10.2`) to the `show_tech.log` file in your working directory, use the following commands:

```
working-directory script show_tech.log
Script started, file is show_tech.log
working-directory telnet 10.10.10.2
Trying 10.10.10.2...
Connected to isp-224.
Escape character is '^]'.

isp-224
login: admin
Password:
[local]isp-224#
[local]isp-224#term len 0
[local]isp-224#show tech-support
```

4. When the command has completed and the CLI prompt appears again, enter the **exit** command twice (to exit the router and the script). The script completes with a message; in this case, `Script done, file is show_tech.log`.

For information about collecting data for troubleshooting, see *Data Collection Guideline for the SmartEdge Router* or

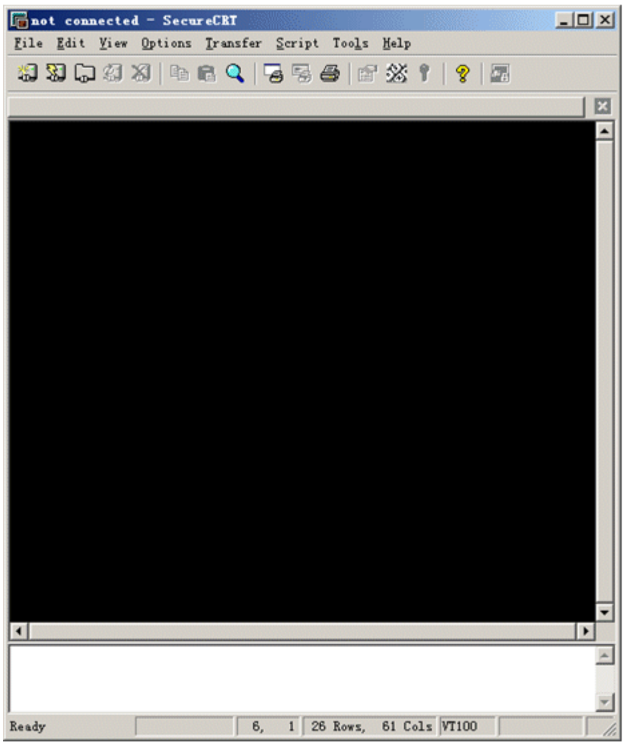
2.2 Enabling Terminal Emulator Logging

Enabling logging of your CLI session by a terminal emulator such as secureCRT or PuTTY is useful for performing offline analysis and providing information for further escalation. Most logon software now supports automatic logging.

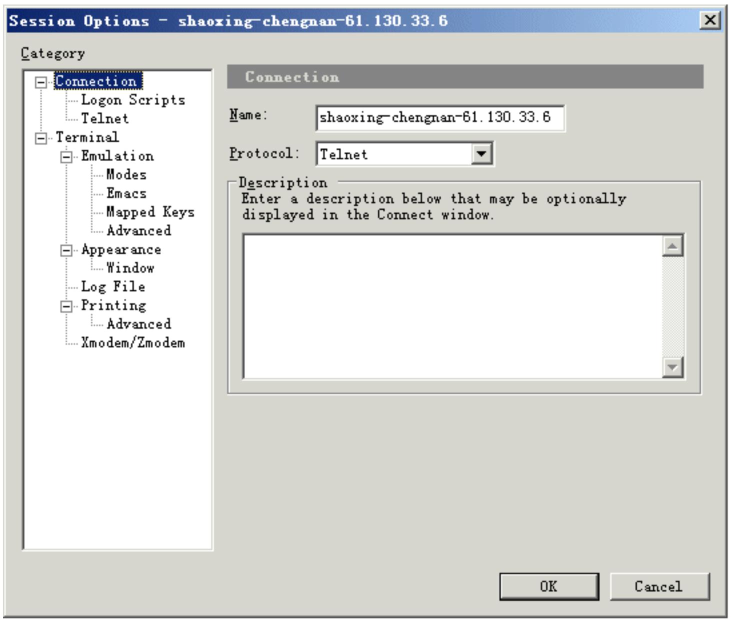
The following procedure shows how to configure secureCRT software to enable automatic logging. (Different versions of secureCRT may require different steps.)

To configure login information and enable automatic logging in secureCRT software:

1. Start the secureCRT software and click **File** > **Connect**.


not connected - SecureCRT
File Edit View Options Transfer Script Tools Help
Ready
6, 1
26 Rows, 61 Cols
VT100

2. Click **new session** and then **Connection**.



Session Options - shaoxing-chengnan-61.130.33.6

Connection

Name: shaoxing-chengnan-61.130.33.6

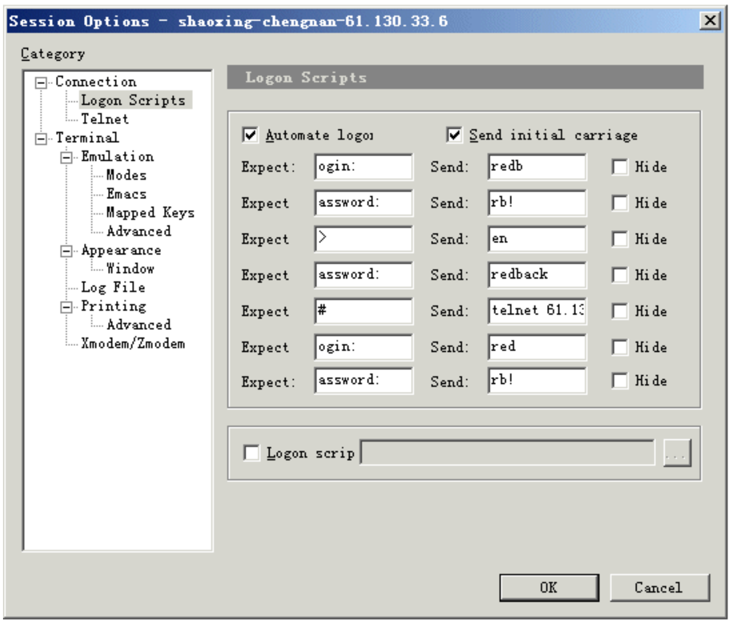
Protocol: Telnet

Description
Enter a description below that may be optionally displayed in the Connect window.

3. In the **Connection** screen **Name** field, type a meaningful name. We recommend that you use the site name concatenated with the node IP address, such as ***`site-name`*** `-61.130.33.6`.

   In the **Protocol** field, type the logon method; for example **`Telnet`**.

4. Click **Logon Scripts** and specify the logon sequences.



☑ Automate logo ☑ Send initial carriage

Expect:	ogin:	Send:	redb	□ Hide
Expect	assword:	Send:	rb!	□ Hide
Expect	>	Send:	en	□ Hide
Expect	assword:	Send:	redback	□ Hide
Expect	#	Send:	telnet 61.13	□ Hide
Expect	ogin:	Send:	red	□ Hide
Expect:	assword:	Send:	rb!	□ Hide

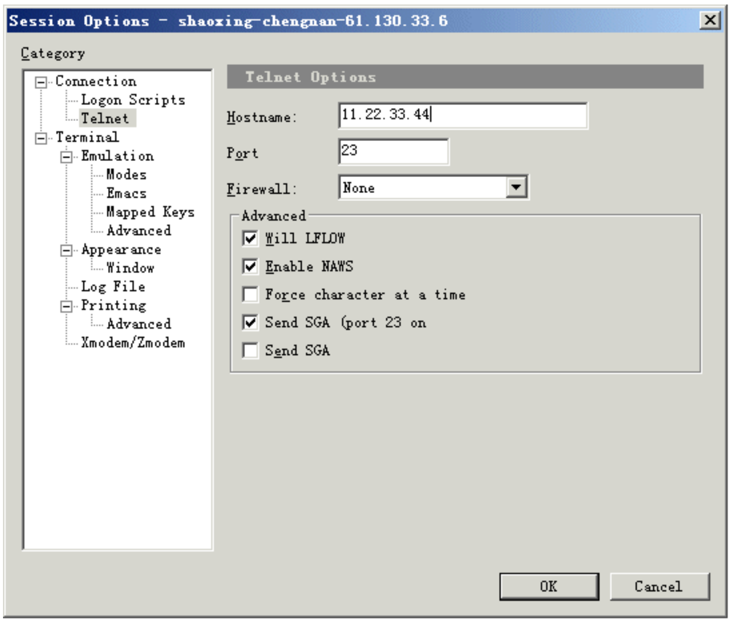
□ Logon scrip

5. Some nodes are protected by one or two intermediate jumphosts for enhanced security. Click **Automate logon** and enter the parameters required during the logon procedure.

   Note: Logon automation may take several attempts to configure properly, and may not be possible if user names and passwords change frequently.

6. Click the Telnet tab.

   In the **Hostname** field or **SSH1** or **SSH2** fields, if the login protocol is Secure Shell (SSH), type the IP address for the direct connection. This can be the node IP, or the springboard IP if one is used.



Session Options - shaoxing-chengnan-61.130.33.6

Category

- Connection
  - Logon Scripts
  - Telnet
- Terminal
  - Emulation
    - Modes
    - Emacs
    - Mapped Keys
    - Advanced
  - Appearance
    - Window
  - Log File
  - Printing
    - Advanced
  - Xmodem/Zmodem

Telnet Options

Hostname: 11.22.33.44

Port: 23

Firewall: None

Advanced

☑ Will LFLOW

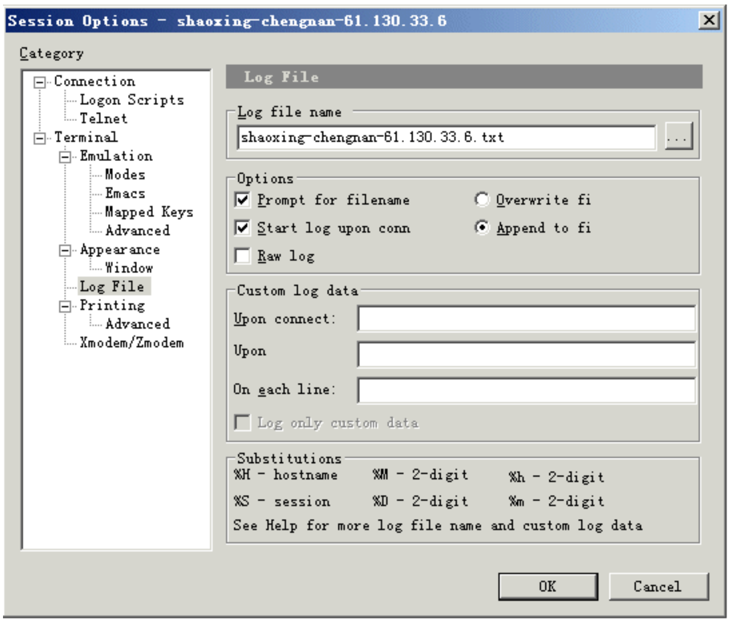
☑ Enable NAWS

☐ Force character at a time

☑ Send SGA (port 23 on

☐ Send SGA

7. For minimum setup, leave the other options at their default settings and click **Log File** .



Session Options - shaoxing-chengnan-61.130.33.6

Category

- Connection
  - Logon Scripts
  - Telnet
- Terminal
  - Emulation
    - Modes
    - Emacs
    - Mapped Keys
    - Advanced
  - Appearance
    - Window
  - Log File
  - Printing
    - Advanced
  - Xmodem/Zmodem

Log File

Log file name: shaoxing-chengnan-61.130.33.6.txt

Options

☑ Prompt for filename ○ Overwrite fi

☑ Start log upon conn ● Append to fi

☐ Raw log

Custom log data

Upon connect:

Upon

On each line:

☐ Log only custom data

Substitutions

```
%H - hostname    %M - 2-digit    %h - 2-digit
%S - session     %D - 2-digit    %m - 2-digit
See Help for more log file name and custom log data
```

8. To predefine a log file for this node, in the **Log File Name** field, type a meaningful name. For flexibility and ease of log file maintenance, especially when you have hundreds of nodes set up, you should use the same name as in the **Name** field on the Connection screen. Each log file is then uniquely associated with a node by its name.

   We recommend selecting **Prompt for filename**, **Start log upon connection**, and **Append to file**.

3 Getting Help for a Command

You can access the online Help for the command-line interface (CLI) in the following ways:

- Use the **`?`** command when issuing a command to display the options available in the command syntax. The system provides two types: full help and partial help. Full help is available when you enter a command argument; for example, **`show ?`** describes each possible argument. Partial help is provided when an abbreviated argument is entered and you want to know what arguments match the input; for example, **`show pr?`**

- Use the **`help`** command to learn how to obtain help. Type the **`help`** command after the prompt. For example: `[local]Redback#`**`help`**

Note: To include the **`?`** character as part of a command when it is not used as a request for online Help, press the **`Esc`** key followed by the **`?`** character.

4 Contexts in Debugging

This section provides information about contexts and displaying debug output.

4.1 How the Active Context Affects Debug Output

The SmartEdge® OS supports multiple contexts. Each context is an instance of a virtual SmartEdge router that runs on the same physical device. A context operates as a separate routing-and-administrative domain with separate routing protocol instances, addressing, authentication, authorization, and accounting. A context does not share this information with other contexts.

There are two types of contexts: local (a system-wide context) and administrator defined (a nonlocal context). The active context (the context that you are in) affects your debug output.

Context-specific debugging refers to navigating to a specific context and running debug commands from it and filtering out all debug output not related to that context. The context-specific output are lines of output identified by a context ID in brackets, which can be displayed either using context-specific debugging or system-wide debugging.

4.1.1 Debugging from the Local Context

To debug all contexts on your router, use the system-wide local context. You see debug output related to this context and all contexts running on the router. For example, to see all OSPF instances on the router, issue the **`debug ospf lsdb`** command in the local context.

```
[local] Redback# debug ospf lsdb
```

When you debug a local context, the software displays debug output for all contexts. When a debug function is context specific, the debug output generated by the local context includes a context ID that you can use to determine the source of the event (the context in which the event has its origin). You can then navigate to the context that contains the event and collect additional information to troubleshoot it.

The following example displays debug output from a local context. The debug output generated by the local context using the **`show debug`** command includes a context ID `0005` (which is highlighted in bold). To find out the source of the debug event (the context name) for context ID `0005,` issue the **`show context all`** command. In the Context ID column look for the Context ID with the last four digits is `0005`—in this case, `0x40080005`, which indicates that the source of the debug event is context `Re-1`.

Note: After a system reboot, context numbers might change.

Debug functions and the **`show context all`** command display context IDs in two different formats: decimal format and hexadecimal, respectively. For example, the debug output displays a context ID in decimal format as `0262`; the **`show context all`** command displays the same ID in hexadecimal format as `0x40080106`.

```
[local]Redback#show debug
OSPF:
   lsdb debugging is turned on
[local]Redback#
Apr 18 12:21:04: %LOG-6-SEC_STANDBY: Apr 18 12:21:04: %CSM-6-PORT:
ethernet 3/7 link state UP, admin is UP
Apr 18 12:21:04: %LOG-6-SEC_STANDBY: Apr 18 12:21:04: %CSM-6-PORT:
ethernet 3/8 link state UP, admin is UP
Apr 18 12:21:05: %CSM-6-PORT: ethernet 3/7 link state UP, admin is UP
Apr 18 12:21:05: %CSM-6-PORT: ethernet 3/8 link state UP, admin is UP
Apr 18 12:21:05: [0002]: %OSPF-7-LSDB: OSPF-1: Area 0.0.0.0 Update
Router LSA 200.1.1.1/200.1.1.1/80000013 cksum 26f1 len 72
Apr 18 12:21:05: [0003]: %OSPF-7-LSDB: OSPF-1: Area 0.0.0.2
Update Router LSA 200.1.2.1/200.1.2.1/80000009 cksum ce79 len 36
Apr 18 12:21:05: [0004]: %OSPF-7-LSDB: OSPF-1: Area 0.0.0.3
Update Sum-Net LSA 0.0.0.0/200.1.3.1/80000001 cksum bb74 len 28
Apr 18 12:21:05: [0004]: %OSPF-7-LSDB: OSPF-1: Area 0.0.0.3
Update Router LSA 200.1.3.1/200.1.3.1/8000000a cksum 142 len 36
Apr 18 12:21:05: [0004]: %OSPF-7-LSDB: OSPF-1: Area 0.0.0.0 Update
Router LSA 200.1.1.1/200.1.1.1/80000013 cksum 26f1 len 72
Apr 18 12:21:05: [0003]: %OSPF-7-LSDB: OSPF-1: Area 0.0.0.0 Update
Router LSA 200.1.1.1/200.1.1.1/80000013 cksum 26f1 len 72
Apr 18 12:21:06 [0005]: %OSPF-7-LSDB: OSPF-1: Area 0.0.0.0 Update
//Associated with Context ID 0x40080005. This is context specific output,
in this case, context Re-1
---------------------------------------------------------------
[local]Redback# show context all
Context Name      Context ID        VPN-RD         Description
----------------------------------------------------------------
local              0x40080001
Rb-1               0x40080002
Rb-2               0x40080003
Rb-3               0x40080004
Re-1               0x40080005 // The source of the debug event for
Re-2               0x40080006 // Context ID 0005 is context Re-1.
Re-3               0x40080007
[local]Redback#
```

4.1.2 Debugging from a Specific Context

The current context affects the output of some debug commands. For example, the **`debug ospf lsdb`** command can be context specific because multiple contexts can exist, each running its own protocols. In this example, you see only the OSPF debug output from context `MyService`. If you run the same command from the local context, you see output from all contexts that have OSPF enabled. The context ID in the debug message logs shows all the contexts for which this debug event is applicable. To debug a specific context for OSPF, navigate to that context—in this example, `MyService`.

```
[local]Redback# context MyService
[MyService] Ericsson# terminal monitor
[MyService] Ericsson# debug ospf lsdb
OSPF:
lsdb debugging is turned on
[MyService]Ericsson#
Feb 27 15:11:24: [0001]: %OSPF-7-LSDB: OSPF-1: Area 0.0.0.0 Update
Router LSA 1.1.1.1/1.1.1.1/8000000c cksum ba60 len 36
Feb 27 15:11:24: [0001]: %OSPF-7-LSDB: OSPF-1: Delete
Net:192.1.1.1[1.1.1.1] Area: 0.0.0.0
Feb 27 15:11:24: [0001]: %OSPF-7-LSDB: OSPF-1: Area 0.0.0.0 Update
Router LSA 1.1.1.1/1.1.1.1/8000000d cksum b861 len 36
Feb 27 15:12:09: [0001]: %OSPF-7-LSDB: OSPF-1: Area 0.0.0.0 Update Net
LSA 192.1.1.1/1.1.1.1/80000002 cksum 1b4a len 32
Feb 27 15:12:09: [0001]: %OSPF-7-LSDB: OSPF-1: Delete
Net:192.1.1.1[1.1.1.1] Area: 0.0.0.0
Feb 27 15:12:09: [0001]: %OSPF-7-LSDB: OSPF-1: Area 0.0.0.0 Update
Router LSA 2.2.2.2/2.2.2.2/80000005 cksum 6ec8 len 36
Feb 27 15:12:09: [0001]: %OSPF-7-LSDB: OSPF-1: Area 0.0.0.0 Update
Router LSA 1.1.1.1/1.1.1.1/80000010 cksum 4f30 len 48
Feb 27 15:12:09: [0001]: %OSPF-7-LSDB: OSPF-1: Area 0.0.0.0 Update Net
LSA 192.1.1.1/1.1.1.1/80000003 cksum 194b len 32
Feb 27 15:12:14: [0001]: %OSPF-7-LSDB: OSPF-1: Area 0.0.0.0 Update
Router LSA 2.2.2.2/2.2.2.2/80000006 cksum 237a len 48
```

4.2 Identifying Context-Specific Debug Functions

Debug functions are either context specific or system wide. For example, the **`debug aaa authen`** command is system wide because negotiation takes place at the port or circuit level, and it is not associated with a context. When you debug from the local context, you see debug output from all contexts. If you debug a function from the local context, use the context ID to determine the source of the debug event (the context that the event is coming from). When you debug from a nonlocal context, you see output only from that context. You can perform context-specific debugging from the local context or from one of the contexts you have configured.

The following three examples show you how to recognize whether a debug function is context specific or applies to the local context. In the first example, context `NiceService` contains context identifier `0002`, which indicates that the **`debug aaa author`** function is context specific.

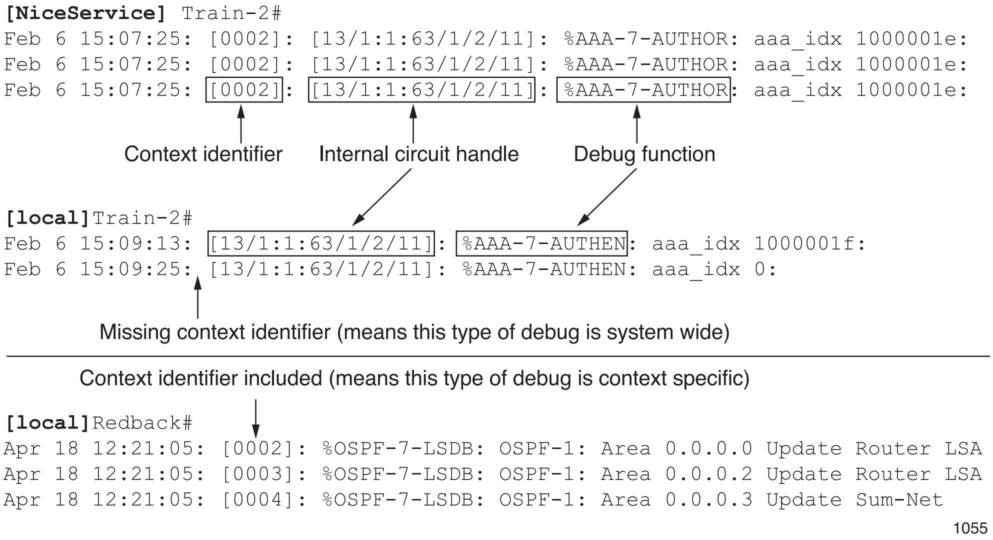
The internal circuit handle, `13/1:1:63/1/2/11`, consists of the following components:

1. **slot/port**—`13/1`

2. **channel: subchannel**—Identifies an individual circuit on a TDM port. `13/1:1:63`is an ATM circuit.

3. Authority (the application that made the circuit, in this case, ATM) `1`, level of circuit (in this example, a traffic bearing Layer 2 circuit) `2`, and the internal ID (a sequential uniquely assigned number) `11`.

In the second example, the **`debug aaa authen`** function in the local context is system wide because no context identifiers are displayed in the show debug output. In the third example, the local context displays context identifiers, `0002`, `0003, and 0004`, which indicates that the source of the LSA updates are context specific. When you issue the **`show context all`** command, these contexts are displayed as `0x40080002` , `0x40080003`, and `0x40080004`.



```
[NiceService] Train-2#
Feb 6 15:07:25: [0002]: [13/1:1:63/1/2/11]: %AAA-7-AUTHOR: aaa_idx 1000001e:
Feb 6 15:07:25: [0002]: [13/1:1:63/1/2/11]: %AAA-7-AUTHOR: aaa_idx 1000001e:
Feb 6 15:07:25: [0002]: [13/1:1:63/1/2/11]: %AAA-7-AUTHOR: aaa_idx 1000001e:

[local]Train-2#
Feb 6 15:09:13: [13/1:1:63/1/2/11]: %AAA-7-AUTHEN: aaa_idx 1000001f:
Feb 6 15:09:25: [13/1:1:63/1/2/11]: %AAA-7-AUTHEN: aaa_idx 0:

[local]Redback#
Apr 18 12:21:05: [0002]: %OSPF-7-LSDB: OSPF-1: Area 0.0.0.0 Update Router LSA
Apr 18 12:21:05: [0003]: %OSPF-7-LSDB: OSPF-1: Area 0.0.0.2 Update Router LSA
Apr 18 12:21:05: [0004]: %OSPF-7-LSDB: OSPF-1: Area 0.0.0.3 Update Sum-Net
```

Context identifier
Internal circuit handle
Debug function
Missing context identifier (means this type of debug is system wide)
Context identifier included (means this type of debug is context specific)
1055

4.3 Displaying Debug Output through the Craft Port

Use the **`logging console`** command (in context configuration mode) to view event log messages on the console. By default this is enabled in the local context.

```
[local]Redback#config
Enter configuration commands, one per line, 'end' to exit
[local]Redback(config)#context local
[local]Redback(config-ctx)#logging console
```

4.4 Displaying Debug Output Through Telnet or SSH

Use the **`terminal monitor`** command (in exec mode) to view event log messages on your terminal when you are connected through Telnet or SSH. To pause debug output at your terminal, type **`CTRL-S`**; to continue, type **`CTRL-C`**.

```
[local]Redback# terminal monitor
```

4.5 Debugging Tasks

Use Table 1 as a guide to troubleshooting subscriber connectivity issues. Check each task that you have completed and document your results. Before you begin, get a description of the problem from the customer and check if the customer has made any recent changes or upgrades to their network.

Table 1 Tasks to Troubleshoot Subscriber Connectivity Issues

#	Task	Command	Notes	Checked ?
1	Display a list of all virtual router instances running on the SmartEdge router.	`show context all`	Issue this command in the local context. Use this command to determine the context from which the debug message is coming. To capture output for all contexts, use the local context.	
2	Determine if the debug function is context specific or system wide.		Consider the type of debug function and how it relates to a context before you troubleshoot an issue. For information about how to recognize the types of debug functions, see Identifying Context-Specific Debug Functions.	
3	Navigate to the context you want to debug.			

Table 1   Tasks to Troubleshoot Subscriber Connectivity Issues

#	Task	Command	Notes	Checked ?

4	Verify that your are in the correct context.	`show context`
		`terminal monitor`
		`show terminal`
			After you navigate to the context that you are debugging, verify that you are in the correct context using the `show context` command.
			If you debug a context and you are not seeing the expected results, you might be in the incorrect context or the `terminal monitor` command might not be enabled in that context when you telnet or ssh to the node.
			Verify that the terminal monitor is enabled by using the `show terminal` command.	

5	Enable the logger process to display debug output.	`logging console`
		`terminal monitor`
		`show terminal`
			For more information about how to display debug output, see Displaying Debug Output through the Craft Port.	

6	Debug the context.	`debug`		
7	Display the debug options that are currently enabled.	`show debug`	You can filter the results by using the `grep` commands. For more information on the grep command options, see the GNU grep documentation available at http://www.gnu.org. Document your results and save your output.	
8	Disable the generation of debug messages.	`no debug`		

5 Using Basic Troubleshooting Commands

For more information about these commands, see the *Command List*. For information about how to Troubleshoot BRAS, see the *BRAS Troubleshooting Guide*.

Table 2 Basic Troubleshooting Commands

Command	Function
`ping`	Test whether the host is reachable.
`ping ancp`	Test DSL circuits by sending a port-management message for Access Node Control Protocol (ANCP) General Switch Management Protocol (GSMP) to the ANCP neighbor peer to test the peer.
`ping arp`	Resolve the MAC address and detects duplicate IP addresses in the system.
`ping atm`	Test ATM PVCs by sending operation, administration, and maintenance (OAM) loopback cells.
`ping cpe`	Resolve the MAC address, detect duplicate IP addresses in the system, and test the data path by sending ICMP echo requests to the CPE.
`ping mpls ldp`	Initiate a MPLS ping across a LDP LSP. For more
`ping mpls mac-address`	Initiate a MPLS ping or a trace to a MAC address in a VPLS network.
`ping mpls pw`	Test the status of a pseudowire.
`ping mpls rsvp`	Initiate a MPLS ping across a RSVP LSP.
`show bindings summary`	Display only summary information for the specified PVCs on the system.
`show chassis`	Display chassis information and the cards that are installed and configured.
`show context all`	Display a list of configured contexts.
`show crashfiles`	Displays the size, location, and name of any crash files located on the system.
`show diag pod`	For any SmartEdge chassis except the SmartEdge 100 chassis, displays the results of the power-on diagnostic (POD) tests.
`show hardware`	Display information about the system hardware. For more information about troubleshooting hardware and how to interpret alarms, see the appropriate SmartEdge router hardware guide.

Table 2 Basic Troubleshooting Commands

Command	Function
`show history`	Displays the command history for the current session
`show ip route summary`	Display summary information for all IP routes.
`show log`	Display information about system event logs or a previously saved log file.
`show memory`	Display statistics about the available and allocated memory in the system memory partition, which is useful for determining if the system is running low on available memory.
`show port`	Display a list of ports that are present or configured in the system.
`show process`	Display the current status of one or all processes running on the system
`show redundancy`	Display the state of the standby controller card and verifies whether it is ready to become active.
`show rmon`	Display RMON information.
`show subscribers summary all`	Display IP information associated with subscribers.
`show system alarm`	Display system-level, card-level, port-level, channel-level, or subchannel-level alarms.
`traceroute`	Trace the IP route that packets take when traveling to the specified destination.
`traceroute mpls`	Initiate a MPLS trace across a RSVP LSP or a LDP LSP.

6 Accessing the SmartEdge System Components

To perform the tasks in this document, you may need to access the SmartEdge system components on the primary and secondary XCRP controller cards.

Table 3 Tasks to Access the SmartEdge System Components

Task	Command
Access Primary and Secondary XCRP Controller Cards	**`telnet`**[1]
Log on to the Standby XCRP card	**`telnet`**[1]
Access the NetBSD Shell Mode	**`start shell`**
Access Open Firmware (OpenBoot) Mode	**`reload`** **`*se`**

[1] See Table 5 or Table 6 for telnet command IP addresses, depending on the SmartEdge platform.

Each controller card runs the NetBSD and VxWorks operating systems, each located on a dedicated compact flash (CF) card and run on a dedicated processor in the SmartEdge 400 or 800 platform (PowerPC) or on a dedicated core in the multicore processor environment on the SmartEdge 1200/1200H platform (MIPS).

NetBSD is the OS on which the SmartEdge OS runs. You may be asked by your support representative to access the NetBSD OS to perform such tasks as reloading NetBSD processes and generating core dumps of their memory at the time of a failure.

VxWorks is the OS that is responsible for most low-level processing, such as driving or monitoring traffic cards.

6.1 Access Primary and Secondary XCRP Controller Cards

Table 4 describes the XCRP controller card terms used in this document.

Table 4 Primary and Secondary XCRP Controller Cards

Term	Description
Primary controller card	XCRP Controller card installed in Slot 7 on a SmartEdge 800 router and slot 6 on a SmartEdge 400 router

Table 4 Primary and Secondary XCRP Controller Cards

Term	**Description**
Secondary controller card	XCRP Controller card installed in Slot 8 on a SmartEdge 800 router and slot 5 on a SmartEdge 400 router
Active controller card	Controller that is currently active or working
Standby controller card	Controller that is currently in standby mode

To enable access to controller cards from the CLI, the SmartEdge OS provides default addresses (IP addresses and ports) for each controller card; for the default slots, IP addresses, and ports for the SmartEdge 400 platform, see Table 5; for the SmartEdge 600, 800, 1200, or 1200H platform, see Table 6.

Table 5 SmartEdge 400 Slots, IP Addresses, and Ports

SmartEdge 400 Slot	IP Address and Port	Destination
XCRP 5	127.0.2.6 23	SmartEdge OS CLI
XCRP 6	127.0.2.5 23	SmartEdge OS CLI

Note: The XCRP that comes up first in slot 5 or slot 6 on a SmartEdge 400 chassis is the primary, active XCRP.

Table 6 SmartEdge 600, 800, 1200, and 1200H Slots, IP Addresses and Ports for Telnet

SmartEdge 600, 800, 1200, and 1200H Slot	IP Address and Port	Destination
XCRP 7	127.0.2.5 23	SmartEdge OS CLI
XCRP 8	127.0.2.6 23	SmartEdge OS CLI

Note: The XCRP that comes up first in slot 7 or slot 8 on a SmartEdge 600, 800, 1200, or 1200H chassis is the primary, active XCRP.

Note: Descriptions and output examples of most commands in this document are based on commands entered on the active controller card; unless noted, the commands also apply to the backup controller card.

6.2 Logging On to the Standby XCRP Controller Card

To collect information or to perform recovery tasks on the standby controller card, log on to it from the active controller card and use the same commands that you would on the active one. The following example shows how to log on to the standby controller card from the active one, assuming that slot 8 contains the active controller card and slot 7 contains the standby one. The `standby` prompt indicates that you are now working on the standby controller.

```
[local]SmartEdge#show chassis | include xcrp
7 : xcrp     7 : xcrp      Yes B
8 : xcrp     8 : xcrp      Yes A

[local]SmartEdge#telnet 127.0.2.5
Trying 127.0.2.5...
Connected to 127.0.2.5
Escape character is '^]

SmartEdge login:the same login name as with active XCRP
Password:the same password as with active XCRP
[local]standby#
```

6.3 Accessing NetBSD Shell Mode

To access the NetBSD OS level from the SmartEdge CLI, use the following command in exec mode:

```
[local]SmartEdge#start shell
```

```
#
```

The # prompt indicates you are now at the NetBSD OS level.

6.4 Accessing and Using Open Firmware (OFW or OpenBoot) Mode

To access the Open Firmware shell (also known as the BootROM or OK mode or NetBSD shell) through the console port on the front of each controller card:

1. Enter the **reload** command (in exec mode) from the console port.

2. Watch the reload progress messages carefully. When the following message appears, type **se*** within five seconds:

   ```
   Auto-boot in 5 seconds - press se* to abort, ENTER to
   boot:
   ```

3. If you typed **se*** within 5 seconds, the OpenBoot `ok` prompt appears. The system sets the autoboot time limit to 5 seconds; however, during some operations, such as a release upgrade, the system sets the time limit to 1 second to speed up the process, then returns it to 5 seconds when the system reboots. (If you missed the time limit, the reload continues; start again with Step 1)

6.4.1 Using the Boot ROM Shell to Verify and Reconfigure Internal Boot Variables

See Section 6.4.1.4 on page 25 for useful Boot ROM commands.

6.4.1.1 Verify and Set Boot Devices

Perform the following steps to set the boot parameters:

1. Access the boot ROM interface (that is, navigate to the `ok` prompt); see Section 6.3 on page 21.

2. Display and verify boot parameters; enter the following boot ROM command:

   ```
   ok printenv
   ```

   This command prints all boot parameters.

3. Verify that the parameters listed in Table 7 are set to the required values.

Table 7 Boot Parameter Values

Parameter	Value
`boot-device`	*flash*
`boot-command`	*bootsys*

4. If these values are not correctly set, enter the following boot ROM command to set the required values:

   ```
   ok setenv parameter value
   ```

6.4.1.2 Verify and Change the Auto-Boot Variable

Note: The `auto-boot?` is set to *true* by default. These instructions are included to correct or change the settings if they have been reset.

The operating system ensures that the standby controller card is synchronized with the active controller card so that, if the active controller card fails, the standby can become active immediately. The operating system occasionally reloads the standby controller card automatically. This requires that the `auto-boot?` variable be set to `true`.

To verify the setting of the `auto-boot?` variable on both controller cards, perform the following tasks:

6.4.1.2.1 Connect a Console to the Console Port on Each Controller Card

The console port is labeled “Craft 2” on the front panel of the controller card. (Two cables are shipped with the system for connecting consoles to the console ports.)

6.4.1.2.2 Set the auto-boot? Variable on the Active Controller Card

To set the `auto-boot?` variable, perform the following steps:

1. Access the Boot ROM interface (OK prompt); see Section 6.3 on page 21.
2. Determine the state of the boot ROM variable, `auto-boot?`:

    - If you see the following message on the console connected to the active controller card, the `auto-boot?` variable has already been set to *true*:

      ```
      Auto-boot in 5 seconds - press se* to abort,
      ENTER to boot:
      ```

3. Enter **`se*`** to cancel the reload process and access the boot loader interface. Proceed to the Set the auto-boot? Variable on the Standby Controller Card section.

    If the message does not appear, the `auto-boot?` variable is set to `false` and the OK prompt appears. Continue with Step 4.

4. Set the `auto-boot?` variable to *true*; enter the following command:

    ```
    ok setenv auto-boot? true auto-boot? = true
    ```

5. Because you have modified the boot loader, enter the following command:

    ```
    ok reset
    ```

    The **`reset`** command resets the hardware and initiates a system reload.

6. Proceed to the Set the auto-boot? Variable on the Standby Controller Card section.

6.4.1.2.3 Set the auto-boot? Variable on the Standby Controller Card

After you are connected to the SmartEdge router through the console that is connected to the standby controller card, perform the following steps:

1. Determine the state of the `auto-boot?` variable

    - If you see the `standby#` prompt on the console, the `auto-boot?` variable is set to *true* . No further action is needed.
    - If you see the `ok` prompt on the console connected to the standby controller card, the `auto-boot?` variable is set to *false*.

2. Set the `auto-boot?` variable to *true*; enter the following command:

    ```
    ok setenv auto-boot? true  auto-boot? = true
    ```

3. Because you have modified the boot loader, enter the following command:

```
ok reset
```

If the active controller card fails, the system will continue to operate with the standby controller card.

6.4.1.3 View and Set the Boot Variables

The boot ROM image is stored in the EEPROM on a controller card.

Table 8 lists the most commonly required environmental variables (for example, needed to download new software images); variables described are configured through the **setenv** boot ROM command.

Table 8 Most Commonly Required Environmental Variables

Variable	Description	Example
ip-addr	IP address and network mask of the Ethernet management port on the controller card to which you will be connected; format is A.B.C.D:E.F.G.H.	155.53.53.254:255.255.252.0
gateway-ip-addr	IP address of the gateway router to the IP network on which the server is located. This address is not used if the server is on the same subnet as the SmartEdge router.	155.53.55.254
server-ip-addr	IP address of the server.	10.21.6.200

To view and set the environmental variables, perform the following steps:

1. Access the boot ROM interface (that is, get to the `ok` prompt):

2. Verify that the arguments listed in Table 8 are set to the required values; enter the **printenv** command; you can specify the variable you want to display, for example:

   ```
   ok printenv ip-addr
   ```

   ```
   ok printenv gateway-ip-addr
   ```

   ```
   ok printenv server-ip-addr
   ```

   If you enter the command with no argument, all environmental variable are displayed.

3. If the values for the arguments listed in Table 8 are not correct, enter the corresponding commands with the correct values:

   ```
   ok setenv ip-addrip-addr
   ```

   ```
   ok setenv gateway-ip-addr gateway-ip-addr
   ```

```
ok setenv server-ip-addr server-ip-addr
```

6.4.1.4 Useful Boot ROM Commands

Table 9 describes the boot ROM commands supported on the SmartEdge router.

Table 9 Boot ROM Commands

Syntax	Description	Argument Values

`boot net` *`filename`*
or
`boot hd:a /flash/`*`filename`*
	Boots from a TFTP server or from /flash.	Name of the file to be booted.

`bootsys`	Loads and runs the software image.	None.
`cd` *`path`*	Changes the currently open device to the one specified by the *`path`* argument. The device also parses any optional arguments to configure the device.	*`path`*—Actual or relative device path. Entering `..` specifies the parent of the current device.
`dev` *`path`*	Changes the currently open device to the one specified by the *`path`* argument. The device also parses any optional arguments to configure the device.	*`path`*—Actual or relative device path. Entering`...` specifies the parent of the current device.
`devalias` [*`name string`*]	Displays all device aliases, if no arguments specified; otherwise, creates a device alias called *`name`* with the value *`string`*.	*`name`*—Optional. Name of the device alias. *`string`*—Optional. String to which the alias refers.
`installsys`	Starts the process of installing software from the boot ROM shell. Can be used to reinstall software on a router that is stuck at the Boot ROM shell (OK prompt).	None.

`load net` *`filename`*
or
`load hd:a /flash/`*`filename`*
	Loads an image from a TFTP server or from /flash.	Name of the file to be downloaded.

Table 9 Boot ROM Commands

Syntax	Description	Argument Values
`printenv [parameter]`	Displays all parameter variables, their current values, and their default values if no argument is specified; otherwise, displays the specified parameter variable.	See Table 10.
`probe-all`	Probes the system for all devices and builds the device tree.	None.
`reset`	Resets the hardware and boots the system.	None.
`set-default var`	Sets the value of a parameter to the default value.	See Table 10.
`set-defaults`	Sets all parameter values back to their default values.	None.
`setenv parameter value`	Sets the value of a parameter to a specified value. All characters entered up to the end of the line (including spaces), are stored in NVRAM.	See Table 10.
`show-devs`	Displays all the devices in the device tree.	None.
`sysinfo`	Displays system information including router, chassis, memory, and environmental information.	None.
`update-bootrom`	Updates and runs the currently loaded boot ROM image.	None.

Table 10 lists the parameters used with these boot ROM commands: **printenv**, **setenv**, and **set-default**.

Table 10 Parameters Used with printenv, set-default, and setenv Boot ROM Commands

Syntax	Description	Values
`auto-boot?`	If true, runs the word in **boot-command** after the standard bootup process; otherwise, when booting, stops at the OFW `ok` prompt. You must then enter the word in **boot-command**.	True or false; the default value is true. If set to `true`, it is easier to reach the OFW shell when reloading the router.
`auto-boot-timeout`	Maximum amount of time, in milliseconds, that the autoboot process waits for a key press before times out. (This is not a standard OpenFirmware parameter.)	Integer; the default value is 5000.

Table 10 Parameters Used with printenv, set-default, and setenv Boot ROM Commands

Syntax	Description	Values
`boot-command`	Command to boot the system if the `auto-boot?` value is true.	String; the default value is *bootsys*.
`boot-device`	Device to use to load the boot ROM image when the `boot` command is issued. The string is usually an alias to the actual device.	String; the default value is flash.
`boot-file-ppc0`	NetBSD file to load from the boot device when the `boot` command is issued.	String; the default value is /p01/netbsd.
`boot-file-ppc1`	vxWorks file to load from the boot device when the `boot` command is issued.	String; the default value is /p01/vxWorks.gz.
`diag-device`	Device to use to load the diagnostic image if the `diag-switch?` value is true when boot is issued.	String; the default value is net.
`diag-file`	Diagnostic file to load if the `diag-switch?` value is true when boot is issued.	String; the default value is diag.
`diag-switch?`	If true, turns on extended tests and displays more verbose output.	True or false; the default value is false.
`fcode-debug`	Ericsson internal use only; do not modify.	true or false; the default value is false.
`gateway-ip-addr`	IP address of the gateway router to the IP network on which the TFTP server is located.	String.
`ignore-cfgfile`	If true, the bootup configuration file is bypassed the first time the system is reloaded, after which the parameter is set to false. This parameter allows you to bypass the loading of a possibly corrupt configuration file. It is set from the ok prompt.	True or false; the default value is false.
`input-device`	Device path to use for the console input. The `keyboard` value is usually an alias to the actual input device determined in some machine-dependent manner.	String; the default value is keyboard.
`inverse-video`	If true, displays text on the console as black-on-white; otherwise, displays text on the console as white-on-black. (This is not a standard OpenFirmware parameter).	True or false; the default value is true.

Table 10 Parameters Used with printenv, set-default, and setenv Boot ROM Commands

Syntax	Description	Values
`ip-addr`	IP address and network mask of the Ethernet management port on the active controller card in the SmartEdge router; format is ***A.B.C.D:E.F.G.H***.	String.
`little-endian`	Ericsson internal use only; do not modify.	True or false; the default value is false.
`load-base`	Ericsson internal use only; do not modify.	0x05600000.
`mac_addr`	MAC address of the active controller card; should be the actual MAC address for the hardware.	String.
`nvramrc`	Seven device aliases; Ericsson internal use only; do not modify.	String.
`oem-banner`	String to display when the **banner** command is issued, if the **oem-banner?** value is set to true.	String.
`oem-banner?`	If true, display the contents of the **oem-banner** value when the **banner** command is issued.	True or false; the default value is false.
`oem-logo`	Bitmap to display if the **oem-logo?** value is true. The contents can be machine-dependent.	Bitmap: 64 x 64 x 1 (512 bytes).
`oem-logo?`	If true, displays the bitmap in the **oem-logo** value in front of the banner when the **banner** command is issued; otherwise, a default logo (or no logo) displays.	True or false; the default value is false.
`output-device`	Name of the device to use for console. This is usually an alias to the actual device.	String; the default value is screen.
`real-base`	Ericsson internal use only; do not modify.	0x00000000.
`real-mode?`	Ericsson internal use only; do not modify.	True.
`real-size`	Ericsson internal use only; do not modify.	0x00080000.
`screen-#columns`	Number of columns desired for console output. If 0 is specified, the largest allowable number is used, depending on the font used.	Integer; the default value is 80.

Table 10 Parameters Used with printenv, set-default, and setenv Boot ROM Commands

Syntax	Description	Values
`screen-#rows`	Number of rows desired for console output. If 0 is specified, the largest allowable number is used, depending on the font used.	Integer; the default value is 24.
`secondary-diag?`	If true, run secondary diagnostics at bootup. (This is not a standard OpenFirmware parameter.)	True or false; the default value is true.
`server-ip-addr`	IP address of the TFTP server.	String.
`update-ofw?`	Ericsson internal use only; do not modify.	False.
`use-nvramc?`	Ericsson internal use only; do not modify.	True.
`user-auth`	If true, prompts users for password authentication. If false, authentication is bypassed. Set this flag to false if password is forgotten or lost. If set to false, it is set to true the next time the device is rebooted.	True or false; the default value is true.
`virt-base`	Ericsson internal use only; do not modify.	Integer; the default value is -1.
`virt-size`	Ericsson internal use only; do not modify.	Integer; the default value is -1.
`vx-config-flags`	Ericsson internal use only; do not modify.	0x0.
`vx-other`	Ericsson internal use only; do not modify.	0x7a.
`vx-target-name`	Ericsson internal use only; do not modify.	String.
`vx-host-name`	Ericsson internal use only; do not modify.	String.

Note: Diagnostic syntax, diag-device, diag-file, and diag-switch?, have no effect on the power-on diagnostics (POD) nor are they affected by the POD.

7 Backing Up Configurations and Data in Memory

Memory storage on the SmartEdge router is on two Compact Flash (CF) cards on the XCRP cards, in three partitions:

- `p01`
- `p02`
- `/flash`, with UNIX-based file systems.

`p01` and `p02` store the OS image files; the active partition stores the most recent image installed, and the standby partition stores the previous image. If they are installed, external CF cards provide mass storage capacity, in two partitions, an`/md` partition and another partition for crash files. If no external CF card is installed on an XCRP card, the `/md` directory is placed on the internal CF card.

For regular file backups, we recommend that you back up the following:

- Crash files to remote location (optionally per context)—with the **`service upload-coredump ftp:`*`url`* `[context` *`ctx-name`*`]`** command (in global configuration mode)
- Log files to a syslog server (per context)—with the **`logging syslog`** ***`ip-addr`*** **`[facility`** ***`sys-fac-name`*`]`** command (in context configuration mode).

Before upgrading the SmartEdge OS software or performing an XCRP switchover, perform the following backups:

1. Save the configuration with one of the following methods:
   - Save the current configuration to **`/flash`** or to a remote location (by FTP or SCP) with the **`save configuration /flash/`*`filename`*** command or the **`save configuration ftp://`*`username@hostname/filename`*** command.
   - Back up the configuration during an upgrade. When the system prompts you to save the current configuration, enter **`y`** and specify the location and filename for the file. If you do not specify them, the SmartEdge OS saves the configuration to `/flash/redback.cfg`.
2. Back up the contents of the `/flash` and `/md` disk partitions by accessing the NetBSD shell and backing up `/flash` and `/md` with the **`ftp`** command. For example, to backup `/flash` to `isp:test@192.168.145.99`, use the following commands:

```
[local]Redback#start shell
#ftp 192.168.145.99
Connected to 192.168.145.99.
220 (vsFTPd 1.2.2)
Name (155.53.12.7:root): isp:test
331 Please specify the password.
Password:password
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> bi
200 Switching to Binary mode.
ftp> ha
Hash mark printing on (1024 bytes/hash mark).
ftp> prompt
Interactive mode off.
ftp> cd backup-directory
250 Directory successfully changed.
ftp> mput *.*
```

When the files are copied, a message such as `226 File receive OK. 1595 bytes sent in 00:00 (1.02 MB/s)` is displayed.

To back up the data on the external CF card, enter the **`bye`** command to exit FTP, switch to the `/md` directory and repeat the process.

After the upgrade, restore the configuration from the location where you saved it with the **`configure`** ***`filename`*** or **`configure ftp:url/`*****`filename`*** commands.