



7705 Service Aggregation Router Gen 2

Release 25.3.R2

Basic System Configuration Guide

3HE 21564 AAAA TQZZA 01

Edition: 01

April 2025

Nokia is committed to diversity and inclusion. We are continuously reviewing our customer documentation and consulting with standards bodies to ensure that terminology is inclusive and aligned with the industry. Our future customer documentation will be updated accordingly.

This document includes Nokia proprietary and confidential information, which may not be distributed or disclosed to any third parties without the prior written consent of Nokia.

This document is intended for use by Nokia's customers ("You"/"Your") in connection with a product purchased or licensed from any company within Nokia Group of Companies. Use this document as agreed. You agree to notify Nokia of any errors you may find in this document; however, should you elect to use this document for any purpose(s) for which it is not intended, You understand and warrant that any determinations You may make or actions You may take will be based upon Your independent judgment and analysis of the content of this document.

Nokia reserves the right to make changes to this document without notice. At all times, the controlling version is the one available on Nokia's site.

No part of this document may be modified.

NO WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF AVAILABILITY, ACCURACY, RELIABILITY, TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, IS MADE IN RELATION TO THE CONTENT OF THIS DOCUMENT. IN NO EVENT WILL NOKIA BE LIABLE FOR ANY DAMAGES, INCLUDING BUT NOT LIMITED TO SPECIAL, DIRECT, INDIRECT, INCIDENTAL OR CONSEQUENTIAL OR ANY LOSSES, SUCH AS BUT NOT LIMITED TO LOSS OF PROFIT, REVENUE, BUSINESS INTERRUPTION, BUSINESS OPPORTUNITY OR DATA THAT MAY ARISE FROM THE USE OF THIS DOCUMENT OR THE INFORMATION IN IT, EVEN IN THE CASE OF ERRORS IN OR OMISSIONS FROM THIS DOCUMENT OR ITS CONTENT.

Copyright and trademark: Nokia is a registered trademark of Nokia Corporation. Other product names mentioned in this document may be trademarks of their respective owners.

© 2025 Nokia.

Table of contents

List of tables.....	10
List of figures.....	11
1 Getting started.....	12
1.1 About this guide.....	12
1.2 Platforms and terminology.....	12
1.3 Conventions.....	13
1.3.1 Precautionary and information messages.....	13
1.3.2 Options or substeps in procedures and sequential workflows.....	13
2 File management.....	15
2.1 SR OS file system.....	15
2.1.1 Storage devices.....	15
2.1.2 URLs.....	16
2.1.3 HTTP digest authentication.....	18
2.1.4 Wildcards and special characters.....	18
2.2 Text editor.....	19
2.2.1 Summary of text editor commands.....	19
2.2.2 Using text editor commands.....	19
2.3 File management tasks in the classic CLI.....	26
2.3.1 Displaying directory and file information.....	27
2.3.2 Modifying file attributes.....	28
2.3.3 Creating directories.....	28
2.3.4 Copying files.....	29
2.3.5 Moving files.....	30
2.3.6 Deleting files and removing directories.....	33
2.3.7 Unzipping files.....	33
2.3.8 Repairing the file system.....	34
2.3.9 Displaying file checksums.....	34
2.4 File management tasks in the MD-CLI.....	34
2.4.1 Displaying directory and file information.....	35
2.4.2 Modifying file attributes.....	36
2.4.3 Creating and navigating directories.....	37

2.4.4	Copying files.....	38
2.4.5	Moving files.....	40
2.4.6	Deleting files and removing directories.....	43
2.4.7	Unzipping files.....	44
2.4.8	Repairing the file system.....	44
2.4.9	Displaying file checksums.....	45
3	System initialization and boot options.....	46
3.1	Boot process.....	46
3.2	Boot Loader.....	48
3.3	Boot Options File.....	48
3.3.1	BOF manual mode.....	48
3.4	Software and configuration.....	49
3.4.1	Management interface configuration modes.....	50
3.5	Initial installation and software update.....	51
3.6	USB recovery boot.....	51
3.7	Storage card content.....	52
3.7.1	7705 SAR-1 storage card content.....	52
3.7.2	ISA and ESA applications.....	54
3.8	Persistent indexes in classic and mixed configuration mode.....	54
3.9	BOF and configuration file encryption.....	55
3.10	System profiles.....	55
3.11	Configuring the Boot Options File with CLI.....	56
3.11.1	Basic BOF configuration.....	56
3.11.2	Common configuration tasks.....	58
3.11.2.1	Searching for the BOF.....	58
3.11.2.2	Accessing the CLI.....	58
3.11.2.3	Console connection.....	58
3.11.2.4	Configuring BOF encryption.....	59
3.11.2.5	Configuring the BOF interactive menu password.....	59
3.11.2.6	Configuring configuration file encryption.....	60
3.11.3	Autoconfigure.....	61
3.11.3.1	Autoconfigure restrictions.....	61
3.11.3.2	DHCP discovery of MAC addresses.....	61
3.11.3.3	IPv6 DUID.....	61
3.11.3.4	IPv6 DHCP RAs.....	62

3.11.4	Service management tasks.....	62
3.11.4.1	System administration commands in the classic CLI.....	62
3.11.4.2	System administration commands in the MD-CLI.....	66
4	Debug configuration.....	71
4.1	Debug configuration in the classic CLI.....	71
4.1.1	Logging debug events in the classic CLI.....	71
4.2	Debug configuration in the MD-CLI.....	72
4.2.1	Logging debug events in the MD-CLI.....	73
4.3	Debug configuration in mixed and model-driven mode.....	74
5	Secure boot.....	76
5.1	Secure Boot chain.....	76
5.2	Operational commands and logs.....	77
5.2.1	Secure Boot state.....	77
5.2.2	Software update.....	78
5.2.3	Update Secure Boot variables.....	78
6	System management.....	79
6.1	System management commands.....	79
6.1.1	System information.....	79
6.1.1.1	Name.....	79
6.1.1.2	Contact.....	79
6.1.1.3	Location.....	79
6.1.1.4	Coordinates.....	80
6.1.1.5	Naming objects.....	80
6.1.1.6	Common language location identifier.....	80
6.1.1.7	DNS security extensions.....	80
6.1.2	System time.....	80
6.1.2.1	Time zones.....	80
6.1.2.2	NTP.....	83
6.1.2.3	CRON.....	84
6.2	IP hashing as an LSR.....	85
6.3	Auto-provisioning.....	85
6.3.1	Auto-provisioning limits.....	87
6.3.2	Auto-provisioning process.....	87

6.3.3	Auto-provisioning DHCP rules.....	87
6.3.4	Auto-provisioning failure.....	88
6.4	Administrative tasks.....	88
6.4.1	Saving configurations.....	89
6.4.2	Specifying post-boot configuration files.....	89
6.4.3	Network timing.....	89
6.4.4	Power supplies.....	89
6.5	System router instances.....	90
6.6	System configuration process overview.....	91
6.7	Configuration notes.....	91
6.8	Configuring system management features.....	91
6.8.1	Saving configurations.....	91
6.9	Basic system configuration.....	92
6.10	Common configuration tasks.....	92
6.10.1	System information.....	92
6.10.1.1	System name.....	93
6.10.1.2	Contact.....	93
6.10.1.3	Location.....	93
6.10.1.4	CLLI code.....	93
6.10.1.5	GPS coordinates.....	93
6.10.2	System time elements.....	94
6.10.2.1	Zone.....	94
6.10.2.2	Summer (daylight saving) time.....	94
6.10.2.3	NTP.....	94
6.10.2.4	SNTP.....	100
6.10.2.5	CRON.....	102
6.10.3	ANCP enhancements.....	103
6.10.4	Configuring backup copies.....	103
6.11	Configuring power supply.....	103
6.12	Configuring multichassis redundancy for LAG.....	104
6.13	Post-boot configuration extension files.....	105
6.13.1	Show command output and console messages.....	105
6.14	Configuring system monitoring thresholds.....	106
6.14.1	Creating events.....	106
6.15	Configuring LLDP.....	107

7	Zero touch provisioning.....	110
7.1	ZTP overview.....	110
7.1.1	Network requirements.....	110
7.1.2	Network support.....	111
7.2	ZTP process overview.....	113
7.2.1	Auto-boot process.....	113
7.2.2	Auto-provisioning process.....	113
7.3	DHCP support for ZTP.....	113
7.3.1	DHCP server offer options.....	114
7.3.1.1	Nokia-specific TLV.....	114
7.3.2	Supported DHCP client options for ZTP.....	114
7.3.3	Supported DHCP server options for ZTP.....	115
7.3.4	DHCP discovery and solicitation.....	115
7.3.4.1	DHCP discovery (IPv4 and IPv6).....	116
7.3.4.2	DHCP solicitation (IPv6).....	116
7.3.5	IPv4 and IPv6 DHCP support.....	117
7.3.5.1	IPv4 route installation details.....	117
7.3.5.2	IPv6 DHCP/RA details.....	117
7.3.5.3	ZTP and DHCP timeouts.....	117
7.4	ZTP procedure details.....	118
7.4.1	Node bootup.....	118
7.4.1.1	Reinitiating ZTP during normal node bootup.....	118
7.4.2	BOF.....	118
7.4.2.1	Compact flash support.....	119
7.4.3	Auto-boot process details.....	119
7.4.3.1	Options and option modification.....	119
7.4.3.2	CLI access.....	120
7.4.3.3	Interrupting auto-boot.....	120
7.4.4	Auto-provisioning process.....	120
7.4.4.1	VLAN discovery.....	121
7.4.4.2	Auto-provisioning procedure.....	121
7.4.4.3	Out-of-band management versus in-band management.....	122
7.4.5	Provisioning files.....	123
7.4.5.1	Provisioning file download.....	123
7.4.5.2	Provisioning file resolution using DNS.....	124

7.4.5.3	File download and redundancy.....	124
7.4.5.4	Configuring the ZTP timeout in the provisioning file.....	124
7.4.5.5	Downloading the image file.....	124
7.4.5.6	Example provisioning file.....	125
7.4.5.7	Proxy support.....	127
7.4.6	Day 0 configuration.....	127
7.4.6.1	Day 0 configuration for multi-slot routers.....	128
7.4.6.2	Day 0 symbols.....	128
7.4.6.3	Sample day 0 configuration template.....	131
7.4.7	Logs and events.....	133
7.5	SZTP.....	133
7.5.1	Staging the secure environment.....	135
7.5.2	Bootstrapping methods.....	135
7.5.3	Installation site process.....	136
7.5.3.1	Initial conveyed information file.....	137
7.5.3.2	Onboarding information.....	139
7.5.3.3	Conveyed information.....	143
8	Standards and protocol support.....	144
8.1	Bidirectional Forwarding Detection (BFD).....	144
8.2	Border Gateway Protocol (BGP).....	144
8.3	Bridging and management.....	145
8.4	Certificate management.....	146
8.5	Ethernet VPN (EVPN).....	146
8.6	gRPC Remote Procedure Calls (gRPC).....	146
8.7	Intermediate System to Intermediate System (IS-IS).....	147
8.8	Internet Protocol (IP) general.....	148
8.9	Internet Protocol (IP) multicast.....	148
8.10	Internet Protocol (IP) version 4.....	149
8.11	Internet Protocol (IP) version 6.....	149
8.12	Internet Protocol Security (IPsec).....	150
8.13	Label Distribution Protocol (LDP).....	151
8.14	Multiprotocol Label Switching (MPLS).....	152
8.15	Network Address Translation (NAT).....	152
8.16	Open Shortest Path First (OSPF).....	152
8.17	Path Computation Element Protocol (PCEP).....	153

8.18	Pseudowire (PW).....	153
8.19	Quality of Service (QoS).....	154
8.20	Remote Authentication Dial In User Service (RADIUS).....	154
8.21	Resource Reservation Protocol - Traffic Engineering (RSVP-TE).....	154
8.22	Routing Information Protocol (RIP).....	155
8.23	Segment Routing (SR).....	155
8.24	Simple Network Management Protocol (SNMP).....	155
8.25	Timing.....	157
8.26	Two-Way Active Measurement Protocol (TWAMP).....	157
8.27	Virtual Private LAN Service (VPLS).....	158
8.28	Yet Another Next Generation (YANG).....	158

List of tables

Table 1: Platforms and terminology.....

12

Table 2: Device names, locations, and support.....

15

Table 3: URL types and syntax.....

16

Table 4: Cutting and pasting or deleting text.....

19

Table 5: Inserting new text.....

20

Table 6: Moving the cursor within the file.....

21

Table 7: Moving the cursor around the screen.....

23

Table 8: Replacing text.....

23

Table 9: Searching for text or characters.....

24

Table 10: Manipulating character and line formatting.....

24

Table 11: Miscellaneous commands.....

25

Table 12: Line editing commands.....

25

Table 13: File command local and remote file system support in the classic CLI.....

26

Table 14: File command local and remote file system support in the MD-CLI.....

35

Table 15: Console configuration parameter values.....

59

Table 16: System-defined time zones and UTC offsets.....

81

Table 17: Supported DHCP client options for ZTP.....

114

Table 18: Supported DHCP server options for ZTP.....

115

Table 19: Supported symbols.....

129

List of figures

Figure 1: Boot process.....

47

Figure 2: BOF manual mode.....

49

Figure 3: Load SR OS configuration.....

50

Figure 4: Files on storage card — 7705 SAR-1.....

53

Figure 5: Secure boot chain of trust.....

77

Figure 6: Example of a network with no DHCP relay.....

86

Figure 7: Example of a network with a DHCP relay.....

86

Figure 8: Example of a network with multiple subnets.....

86

Figure 9: System configuration and implementation flow.....

91

Figure 10: Auto-provisioning with all components in the same subnet.....

111

Figure 11: Auto-provisioning with only file and DHCP servers in the same subnet.....

112

Figure 12: Auto-provisioning with all components in different subnets.....

112

Figure 13: SZTP process.....

134

Figure 14: Installation site SZTP process.....

136

1 Getting started


1.1 About this guide

This guide describes system concepts and provides configuration explanations and examples to configure SR OS boot option file (BOF), file system, and system management functions.

This guide is organized into functional chapters and provides concepts and descriptions of the implementation flow, as well as Command Line Interface (CLI) syntax and command usage.

Unless otherwise indicated, the topics and commands described in this guide apply only to the 7705 SAR Gen 2 platforms listed in [Platforms and terminology](#).


Command outputs shown in this guide are examples only; actual displays may differ depending on supported functionality and user configuration.



Note: Unless otherwise indicated, CLI commands, contexts, and configuration examples in this guide apply for both the classic CLI and the MD-CLI.


The SR OS CLI trees and command descriptions can be found in the following guides:

- *7705 SAR Gen 2 Classic CLI Command Reference Guide*
- *7705 SAR Gen 2 Clear, Monitor, Show, Tools CLI Command Reference Guide* (for both the MD-CLI and classic CLI)
- *7705 SAR Gen 2 MD-CLI Command Reference Guide*



Note: This guide generically covers Release 25.x.Rx content and may contain some content that will be released in later maintenance loads. See the *SR OS R25.x.Rx Software Release Notes*, part number 3HE 21562 000x TQZZA, for information about features supported in each load of the Release 25.x.Rx software. For a list of features and CLI commands that are present in SR OS but not supported on the 7705 SAR Gen 2 platforms, see "SR OS Features not Supported on SAR Gen 2" in the *SR OS R25.x.Rx Software Release Notes*.

1.2 Platforms and terminology



Note: Unless explicitly noted otherwise, this guide uses the terminology defined in the following table to collectively designate the specified platforms.

Table 1: Platforms and terminology

Platform	Collective platform designation
7705 SAR-1	7705 SAR Gen 2

1.3 Conventions

This section describes the general conventions used in this guide.

1.3.1 Precautionary and information messages

The following information symbols are used in the documentation.



DANGER: Danger warns that the described activity or situation may result in serious personal injury or death. An electric shock hazard could exist. Before you begin work on this equipment, be aware of hazards involving electrical circuitry, be familiar with networking environments, and implement accident prevention procedures.



WARNING: Warning indicates that the described activity or situation may, or will, cause equipment damage, serious performance problems, or loss of data.



Caution: Caution indicates that the described activity or situation may reduce your component or system performance.



Note: Note provides additional operational information.



Tip: Tip provides suggestions for use or best practices.

1.3.2 Options or substeps in procedures and sequential workflows

Options in a procedure or a sequential workflow are indicated by a bulleted list. In the following example, at step 1, the user must perform the described action. At step 2, the user must perform one of the listed options to complete the step.

Example: Options in a procedure

1. User must perform this step.
2. This step offers three options. User must perform one option to complete this step.
 - This is one option.
 - This is another option.
 - This is yet another option.

Substeps in a procedure or a sequential workflow are indicated by letters. In the following example, at step 1, the user must perform the described action. At step 2, the user must perform two substeps (a. and b.) to complete the step.

Example: Substeps in a procedure

1. User must perform this step.
2. User must perform all substeps to complete this action.
 - a. This is one substep.

-
- b.** This is another substep.

2 File management

This chapter provides information about file system management.

2.1 SR OS file system

The SR OS file system is used to store files used and generated by the system; for example, image files, configuration files, logging files, and accounting files.

The file commands allow you to copy, create, move, and delete files and directories, navigate to a different directory, and display file or directory contents and the image version.

Although some of the storage devices on routers are not actually compact flash devices all storage devices are referred to as compact flash.

2.1.1 Storage devices

On the 7705 SAR Gen 2, each control processor can have up to two storage devices numbered two and three. The names for these devices are:

- cf2:
- cf3:

The above device names are relative device names as they refer to the devices local to the control processor with the current console session. The colon (":") at the end of the name indicates it is a device.

The 7705 SAR Gen 2 boots from an internal embedded MultiMediaCard (eMMC).

Table 2: Device names, locations, and support

7705 SAR Gen 2 platform	Device name and location	
	cf2:	cf3:
7705 SAR-1	USB port	eMMC

The USB port must be administratively enabled before it can be used. Use the following command to enable the USB port:

- **MD-CLI**

```
configure system usb cf2 admin-state enable
```

- **classic CLI**

```
configure system usb cf2 no shutdown
```



Note: On the 7705 SAR-1, only the USB on cf2: is removable.



Note: To prevent corrupting open files in the file system, only remove a storage device that is administratively shutdown. SR OS gracefully closes any open files on the device, so it can be safely removed.

2.1.2 URLs

The arguments for the SR OS file commands are modeled after standard universal resource locator (URL). A URL refers to a file (a *file-url*) or a directory (a *directory-url*).

SR OS supports operations on both the local file system and on remote files. For the purposes of categorizing the applicability of commands to local and remote file operations, URLs are divided into five types of URLs: **local**, **ftp**, **tftp**, **http**, **https**, and **scp**. The syntax for each of the URL types are listed in [Table 3: URL types and syntax](#).

Table 3: URL types and syntax

URL type	Syntax	Notes
<i>local-url</i>	<code>[cflash-id:\]path</code>	<i>cflash-id</i> is the compact flash device name. Values: cf2: , cf3:
<i>ftp-url</i>	<code>ftp://[username[:password]@]host/path</code>	An absolute FTP path from the root of the remote file system. <i>username</i> is the FTP username <i>password</i> is the FTP user password <i>host</i> is the FTP server <i>path</i> is the path to the directory or file
	<code>ftp://[username[:password]@]host/.path</code>	A relative FTP path from the user's home directory. Note the period and slash (".") in this syntax compared to the absolute path.
<i>tftp-url</i>	<code>tftp://host[/path]/filename</code>	TFTP is only supported for operations on a <i>file-url</i> .
<i>http-url</i>	<code>http://[username[:password]@]host[:port]/path</code>	<i>host</i> is the HTTP server <i>port</i> defaults to 80
<i>https-url</i>	<code>https://[username[:password]@]host[:port]/path</code>	<i>host</i> is the HTTPS server <i>port</i> defaults to 443
<i>scp-url</i>	<code>scp://username @host:path</code>	<i>username</i> is the SSH username <i>host</i> is the SSH server <i>path</i> is the path to the directory or file

If the host portion of a URL is an IPv6 address, enclose the URL in quotes and the address in square brackets, as shown in the following examples.

Example

- `"ftp://username:password@[2001:db8:3333:4444:5555:6666:7777:8888]/testfile.txt"`
- `"scp://username@[2001:db8:3333:4444:5555:6666:7777:8888]/testfile.txt"`

The system accepts forward slash (/) or backslash (\) characters to delimit directory and/or filenames in URLs. Similarly, the SR OS SCP client application can use either slash or backslash characters, but not all SCP clients treat backslash characters as equivalent to slash characters. In particular, UNIX systems often interpret the backslash character as an escape character. This can cause problems when using an external SCP client application to send files to the SCP server. If the external system treats the backslash like an escape character, the backslash delimiter gets stripped by the parser and is not transmitted to the SCP server.

For example, a destination directory specified as `"cf2:\dir1\file1"` is transmitted to the SCP server as `"cf2:dir1file1"` where the backslash escape characters are stripped by the SCP client system before transmission. On systems where the client treats the backslash like an escape character, a double backslash (\\) or the forward slash (/) can typically be used to properly delimit directories and the filename.

When a special character is used in a password, it can cause issues when that password is encoded as part of a URL. To prevent this issue, percent encoding can be used. Percent encoding is a mechanism to encode 8-bit characters that have specific meaning in the context of URLs. The encoding consists of substitution of a percent character (%) followed by the hexadecimal representation of the ASCII value of the replaced character.

Some file manipulation commands such as copying, removing, or moving files, may request access to an HTTP or HTTPS server. If an HTTP or HTTPS server redirects the system to a different URL (from an "HTTP 301" error or similar response), the system prompts the user "y/n" to either repeat the operation with the new URL or terminate it. These file commands can be configured to force the HTTP redirects without prompting or they can be configured to refuse HTTP redirects. If a file command is redirected more than eight times, or if it queries an HTTPS URL and gets redirected to an HTTP URL, the command automatically terminates as a security measure.

Use the following command to refuse HTTP redirects:

- **MD-CLI**

```
copy source-url destination-url direct-http
```

- **classic CLI**

```
copy source-url dest-url no-redirect
```

Use the following command to force the HTTP redirects without prompting:

- **MD-CLI**

```
copy source-url destination-url force
```

- **classic CLI**

```
copy source-url dest-url force
```

When connecting to an HTTPS server, the system verifies the server's TLS certificate. For the certificate to pass verification, the system must have a CA profile already configured for the server's Certificate Authority (CA), which specifies up-to-date certificate and CRL files. HTTPS file commands do not use the Online

Certificate Status Protocol (OCSP). If the certificate was issued by an intermediate CA, the system must have a CA profile for every CA tracing back to the root CA. If the server's certificate fails verification for any reason, the file command terminates. See the *7705 SAR Gen 2 Multiservice ISA and ESA Guide* for more information about CA profiles.

Use the following CLI command to configure the CA profile:

```
configure system security pki ca-profile
```

An HTTPS **file** command may also include a **client-tls-profile** configuration parameter, referring to a client TLS profile that provides the cipher list, client certificate, and trust anchor the system uses when communicating with the HTTPS server. See the *7705 SAR Gen 2 System Management Guide* for more information about client TLS profiles.

A **file** command that connects to an HTTP or HTTPS server outside the local network may need to use an HTTP proxy. The user may specify a proxy server (which must be an HTTP URL).

2.1.3 HTTP digest authentication

For HTTP or HTTPS **file** commands only, the HTTP digest authentication scheme can be used with the HTTP authentication mechanism as described in RFC 7616 and RFC 2617. The following hash algorithms are supported:

- MD5
- SHA-256
- SHA-512/256

2.1.4 Wildcards and special characters

SR OS supports the standard wildcard characters. The asterisk (*) can represent zero or more characters in a string of characters, and the question mark (?) can represent any one character and must be enclosed in quotation marks (" ").

Example: MD-CLI

```
[file "cf3:\"]
A:admin@node-2# copy bof.* testdir
11 file(s) copied.

[file "cf3:\"]
A:admin@node-2#
```

Example: classic CLI

```
A:node-2>file cf3:\ # copy bof.* testdir
Copying file cf3:\bof.cfg-1 ... OK
Copying file cf3:\bof.cfg-2 ... OK
Copying file cf3:\bof.cfg-3 ... OK
Copying file cf3:\bof.cfg-4 ... OK
Copying file cf3:\bof.cfg-5 ... OK
Copying file cf3:\bof.cfg-6 ... OK
Copying file cf3:\bof.cfg-7 ... OK
Copying file cf3:\bof.cfg-8 ... OK
Copying file cf3:\bof.cfg-9 ... OK
```

```
Copying file cf3:\bof.cfg.1 ... OK
Copying file cf3:\bof.cfg ... OK
11 file(s) copied.
A:node-2>file cf3:\ #
```

2.2 Text editor

The text editor allows the user to edit an ASCII text file. When the user modifies the configuration using a configuration mode, the system validates whether the user is allowed to perform configuration changes. When the text editor is used to edit a configuration file, these validation checks do not occur. For this reason, administrator privileges are required to access the text editor, and the user profile must be modified to allow this access.

Access permission for the directory where the file resides must be granted before a user can open, read, or write a file. If the user does not have permission to access the directory, the operation is denied.

2.2.1 Summary of text editor commands

The text editor operates in the following modes:

- **command mode**

In this mode, every character entered is a command that causes an action to be taken on the text file. For example, the character “O” typed in command mode causes the text editor to enter insert mode.

- **insert mode**

In this mode, every character typed is added to the text in the file. Pressing Esc turns off insert mode.

2.2.2 Using text editor commands

Use the commands described in the tables in this section to do the following:

- start and end text editor sessions
- move within a file
- enter new text
- modify, move, and delete existing text
- read from and write to other files

The following table describes the commands to cut, paste, and delete text.

Table 4: Cutting and pasting or deleting text

Text editor command	Description
"	Specify a buffer to be used with any of the commands using buffers. Follow the quotation mark (") character with a letter or a number that corresponds to a buffer.

Text editor command	Description
d	Delete text. The “ dd ” command deletes the current line. A count specifies the number of lines to delete. Deleted text is placed in the buffer specified with the “” command. If no buffer is specified, the general buffer is used.
D	Delete to the end of the line from the current cursor position.
p	Paste the specified buffer after the current cursor position or line. If no buffer is specified using the “” command, the general buffer is used.
P	Paste the specified buffer before the current cursor position or line. If no buffer is specified using the “” command, the general buffer is used.
x	Delete the character under the cursor. A count specifies the number of characters to delete. The characters after the cursor are deleted.
X	Delete the character before the cursor.
y	Yank text and place the result into a buffer. The “ yy ” command yanks the current line. Entering a number yanks the specified number of lines. The buffer can be specified with the “” command. If no buffer is specified, the general buffer is used.
Y	Yank the current line into the specified buffer. If no buffer is specified, the general buffer is used.

The following table describes the commands to insert new text.

Table 5: Inserting new text

Text editor command	Description
A	Append text at the end of the current line.
I	Insert text from the beginning of a line.
O	Enter insert mode in a new line above the current cursor position.
a	Enter insert mode and append text after the current cursor position. A preceding count inserts all the text that was inserted the specified number of times.
i	Enter insert mode and insert typed text before the current cursor position. A preceding count inserts

Text editor command	Description
	all the text that was inserted the specified number of times.
o	Enter insert mode in a new line below the current cursor position.

The following table describes the commands used to move the cursor within the file.

Table 6: Moving the cursor within the file

Text editor command	Description
^B	Scroll backward one page. A count scrolls that many pages.
^D	Scroll forward half a window. A count scrolls that many lines.
^F	Scroll forward one page. A count scrolls that many pages.
^H	Move the cursor one space to the left. A count moves that many spaces.
^J	Move the cursor down one line in the same column. A count moves that many lines down.
^M	Move to the first character on the next line.
^N	Move the cursor down one line in the same column. A count moves that many lines down.
^P	Move the cursor up one line in the same column. A count moves that many lines up.
^U	Scroll backward half a window. A count scrolls that many lines.
\$	Move the cursor to the end of the current line. A count moves to the end of the following lines.
%	Move the cursor to the matching parenthesis or brace.
^	Move the cursor to the first non-whitespace character.
(Move the cursor to the beginning of a sentence.
)	Move the cursor to the beginning of the next sentence.
{	Move the cursor to the preceding paragraph.
}	Move the cursor to the next paragraph.

Text editor command	Description
	Move the cursor to the column specified by the count.
+	Move the cursor to the first non-whitespace character in the next line.
-	Move the cursor to the first non-whitespace character in the previous line.
—	Move the cursor to the first non-whitespace character in the current line.
0	Move the cursor to the first column of the current line.
B	Move the cursor back one word, skipping over punctuation.
E	Move the cursor forward to the end of a word, skipping over punctuation.
G	Go to the line number specified as the count. If no count is specified, go to the end of the file.
H	Move the cursor to the first non-whitespace character at the top of the screen.
L	Move the cursor to the first non-whitespace character at the bottom of the screen.
M	Move the cursor to the first non-whitespace character in the middle of the screen.
W	Move forward to the beginning of a word, skipping over punctuation.
b	Move the cursor back one word. If the cursor is in the middle of a word, move the cursor to the first character of that word.
e	Move the cursor forward one word. If the cursor is in the middle of a word, move the cursor to the last character of that word.
h	Move the cursor one character position to the left.
j	Move the cursor down one line.
k	Move the cursor up one line.
l	Move the cursor one character position to the right.
w	Move the cursor forward one word. If the cursor is in the middle of a word, move the cursor to the first character of the next word.

The following table describes the commands to move the cursor around the screen.

Table 7: Moving the cursor around the screen

Text editor command	Description
^E	Scroll forward one line. A count scrolls that many lines.
^Y	Scroll backward one line. A count scrolls that many lines.
z	<p>Redraw the screen with the following options:</p> <ul style="list-style-type: none"> • z<return> puts the current line on the top of the screen. • z. puts the current line on the center of the screen. • z- puts the current line on the bottom of the screen. <p>If you specify a count before the z command, it changes the current line to the line specified. For example, 16z. puts line 16 on the center of the screen.</p>

The following table describes the commands to replace text.

Table 8: Replacing text

Text editor command	Description
C	Change to the end of the line from the current cursor position.
R	Replace characters on the screen with a set of characters entered, ending with Esc.
S	Change an entire line.
c	The cc command changes the current line. A count changes that many lines.
r	Replace one character under the cursor. Specify a count to replace a number of characters.
s	Substitute one character under the cursor, and enter insert mode. Specify a count to substitute a number of characters. A dollar sign (\$) is placed at the last character to be substituted.

The following table describes the commands to search for text or characters in the file.

Table 9: Searching for text or characters

Text editor command	Description
,	Repeat the last f , F , t or T command in the reverse direction.
/	Search the file forward for the string specified after the forward slash (/).
;	Repeat the last f , F , t or T command.
?	Search the file backward for the string specified after the ?.
F	Search the current line backward for the character specified after the F command. If found, move the cursor to the position.
N	Repeat the last search done by forward slash (/) or question mark (?) in the backward direction.
T	Search the current line backward for the character specified after the T command, and move to the column after the character, if it is found.
f	Search the current line for the character specified after the f command. If found, move the cursor to the position.
n	Repeat the last search done by forward slash (/) or question mark (?) in the forward direction
t	Search the current line for the character specified after the t command, and move to the column before the character, if it is found.

The following table describes the commands to manipulate character and line formatting.

Table 10: Manipulating character and line formatting

Text editor command	Description
~	Switch the case of the character under the cursor.
<	Shift the lines up to the left by one shiftwidth. The << command shifts the current line to the left and can be specified with a count.
>	Shift the lines up to the right by one shiftwidth. The >> command shifts the current line to the right and can be specified with a count.
J	Join the current line with the next one. A count joins that many lines.

The following table describes miscellaneous commands.

Table 11: Miscellaneous commands

Text editor command	Description
^G	Show the current file name and the status.
^L	Clear and redraw the screen.
^R	Redraw the screen removing false lines.
^[Cancel a partially formed command (Esc).
^^	Go back to the last file edited.
&	Repeat the previous :s command.
.	Repeat the last command that modified the file.
:	Begin typing a line editing command. The command is executed when the user presses Enter.
@	Type the command stored in the specified buffer.
U	Restore the current line to the previous state before the cursor entered the line.
m	Mark the current position with the character specified after the m command.
u	Undo the last change to the file. Redo the change by typing u again.
ZZ	Exit the editor, saving if any changes were made.

From the text editor, use the **:** command to enter a line editing command. To modify more than one line using specific commands (such as **:s** and **:w**), the range must be specified before the command. For example, to substitute lines 3 through 15, the command is **:3,15s/from/this/g**.

The following table describes the commonly used line editing commands.

Table 12: Line editing commands

Text editor command	Description
:ab string strings	Abbreviation. If a word is typed in the text editor corresponding to string1, the editor automatically inserts the corresponding words. For example, the abbreviation :ab vprn Virtual Private Routed Network" inserts the words "Virtual Private Routed Network" whenever the word "vprn" is typed.
:map keys new_seq	Map a key or a sequence of keys to another key or sequence of keys.

Text editor command	Description
:q	Quit the text editor. If changes have been made, the editor issues a warning message.
:q!	Quit the text editor without saving changes.
:s/pattern/to_pattern/options	Substitute the specified pattern with the string in the to_pattern . Without options, only the first occurrence of the pattern is substituted. If a g is specified, all occurrences are substituted.
:set [all]	Sets some customizing options. The :set all command displays all options.
:una string	Removes the abbreviation previously defined by :ab .
:unm keys	Removes the mapping defined by :map .
:vi filename	Starts editing a new file. If changes have not been saved, the editor displays a warning message.
:w	Write contents of the current file.
:w filename	Write the contents of the buffer to the file name specified.
:w >> filename	Append the contents of the buffer to the file name.
:wq	Write the contents of the buffer and quit.

2.3 File management tasks in the classic CLI

The following sections are basic system tasks that can be performed in the classic CLI.

For more information about the supported classic CLI commands, see the *7705 SAR Gen 2 Classic CLI Command Reference Guide*.

When a file system operation is performed that can potentially remove or overwrite a file system entry, a prompt appears to confirm the action. The **force** keyword performs the operation without displaying the confirmation prompt.

All the commands can operate on the local file system. The following table indicates the commands that also support remote file operations.

Table 13: File command local and remote file system support in the classic CLI

Command	local-url	ftp-url	tftp-url	http-url	https-url
attrib	✓				
cd	✓	✓			
copy	✓	✓	source only	✓	✓

Command	local-url	ftp-url	tftp-url	http-url	https-url
copy (recursive)	✓	✓			
checksum	✓	✓	✓		
delete	✓	✓		✓	✓
dir	✓	✓			
md	✓	✓			
move	✓	✓		✓	✓
move (recursive)	✓	✓			
rd	✓	✓			
scp	source only				
type	✓	✓	✓	✓	✓
unzip	✓	✓	source only		
version	✓	✓	✓		
vi	✓				

2.3.1 Displaying directory and file information

Use the **dir** command to display a list of files on a file system. The **type** command displays the contents of a file. The **version** command displays the version of an SR OS .tim image file.

Example: Display directory and file information

```
A:node-2>file cfl:\ # dir
Volume in drive cfl on slot A has no label.
Directory of cfl:\
01/01/1980  12:00a                7597 test.cfg
01/01/1980  12:00a                 957 b.
08/19/2001  02:14p            230110 B00TROM.SYS
01/01/1980  12:00a                133 NVRAM.DAT
04/03/2003  05:32a                1709 103.ndx
01/28/2003  05:06a                1341 103.cftg.ndx
01/28/2003  05:06a            20754 103.cftg
04/05/2003  02:20a    <DIR>          test
                    15 File(s)          338240 bytes.
                    3 Dir(s)           1097728 bytes free.
A:node-2>file cfl:\ # type fred.cfg
# Saved to /cflash1/fred.cfg
# Generated THU FEB 21 01:30:09 2002 UTC
exit all
config
#-----
# Chassis Commands
#-----
card 2 card-type faste-tx-32
```

```

exit
#-----
# Interface Commands
#-----
# Physical port configuration
interface faste 2/1
    shutdown
    mode network
exit
interface faste 2/2
    shutdown
exit
interface faste 2/3
    shutdown
exit
interface faste 2/4
A:node-2>file cf1:\ # version boot.tim
TiMOS-L-1.0.B3-8
A:node-2>file cf1:\ #

```

2.3.2 Modifying file attributes

The system administrator can change the read-only attribute in the local file. Enter the **attrib** command with no options to display the contents of the directory and the file attributes.



Note: A file with an "R" preceding the filename indicates that the file is read-only.

Example: File configuration output

```

A:node-2>file cf3:\ # attrib
cf3:\bootlog.txt
cf3:\bof.cfg
cf3:\boot.ldr
cf3:\bootlog_prev.txt
cf3:\B0F.SAV
A:node-2>file cf3:\ # attrib +r B0F.SAV
A:node-2>file cf3:\ # attrib
cf3:\bootlog.txt
cf3:\bof.cfg
cf3:\boot.ldr
cf3:\bootlog_prev.txt
R   cf3:\B0F.SAV

```

2.3.3 Creating directories

Use the **md** command to create a new directory in the local file system, one level at a time.

Enter the **cd** command to navigate to different directories.

Example: Creating three levels of directories

```

A:node-2>file cf1:\ # md test1
A:node-2>file cf1:\test1\ # cd test1
A:node-2>file cf1:\test1\ # md test2
A:node-2>file cf1:\test1\test2\ # cd test2
A:node-2>file cf1:\test1\test2\ # md test3
A:node-2>file cf1:\test1\test2\ # cd test3

```

```
A:node-2>file cf1:\test1\test2\test3 #
```

2.3.4 Copying files

Use the **copy** command to copy files to or from a flash device or an FTP-TFTP server.

The **copy** command supports wildcards.

The **scp** command copies files between hosts on a network. It uses SSH for data transfer, uses the same authentication, and provides the same security as SSH.

The source file for the **scp** command must be local. The file must reside on the router. The destination file does not need to be local, but it must be in the `user@host:file-name` format.

Example: Image file and network host copy

```
A:node-2>file cf1:\ # copy 104.cfg cf1:\test1\test2\test3\test.cfg
A:node-2>file cf1:\ # scp file1 admin@192.168.x.x:cf1:\file1
A:node-2>file cf1:\ # scp file2 user2@192.168.x.x:/user2/file2
A:node-2>file cf1:\ # scp cf2:/file3 admin@192.168.x.x:cf1:\file3
```

Use the **recursive** keyword to recursively copy files and directories. If files or directories already exist, the operator is prompted to overwrite them. When the **force** keyword is enabled, a positive response to the overwrite prompts is assumed. The operator is not prompted for confirmation and the existing files or directories are overwritten.

Example: Recursive directory copy

```
A:node-2# file dir
Volume in drive cf3 on slot A is SR0S VM.

Volume in drive cf3 on slot A is formatted as FAT32

Directory of cf3:\

09/19/2021 06:11p 196 B0F.CFG
09/19/2021 06:11p 0 CONFIG.CFG
09/19/2021 06:11p 101 NVRAM.DAT
09/19/2021 07:22p SUPPORT/
09/19/2021 06:11p SYSLINUX/
09/19/2021 06:11p TIMOS/
09/23/2021 09:03a 27459 bootlog.txt
09/20/2021 09:56a 27326 bootlog_prev.txt
09/23/2021 01:21p 319 nvsys.info
09/21/2021 07:19p recursive3/
09/23/2021 08:22a ssh/
7 File(s) 55402 bytes.
6 Dir(s) 612319232 bytes free.

A:node-2# file dir recursive3

Volume in drive cf3 on slot A is SR0S VM.

Volume in drive cf3 on slot A is formatted as FAT32

Directory of cf3:\recursive3

09/21/2021 07:18p ./
09/21/2021 07:18p ../
09/21/2021 07:19p 7 file1.txt
```

```

09/21/2021 07:19p recursive2/
1 File(s) 7 bytes.
3 Dir(s) 612319232 bytes free.

A:node-2# file copy recursive3 recursive1 recursive
Copying directory cf3:\recursive3
Copying directory cf3:\recursive3\recursive2
Copying file cf3:\recursive3\recursive2\file2.txt ... OK
Copying file cf3:\recursive3\recursive2\file3.txt ... OK
Copying file cf3:\recursive3\file1.txt ... OK
2 dir(s) and 3 file(s) copied.
A:sros# file dir

Volume in drive cf3 on slot A is SROS VM.

Volume in drive cf3 on slot A is formatted as FAT32

Directory of cf3:\

09/19/2021 06:11p 196 B0F.CFG
09/19/2021 06:11p 0 CONFIG.CFG
09/19/2021 06:11p 101 NVRAM.DAT
09/19/2021 07:22p SUPPORT/
09/19/2021 06:11p SYSLINUX/
09/19/2021 06:11p TIMOS/
09/23/2021 09:03a 27459 bootlog.txt
09/20/2021 09:56a 27326 bootlog_prev.txt
09/23/2021 01:21p 319 nvsys.info
09/23/2021 02:42p recursive1/
09/21/2021 07:19p recursive3/
09/23/2021 08:22a ssh/
7 File(s) 55402 bytes.
7 Dir(s) 612298752 bytes free.

A:node-2# file dir recursive1

Volume in drive cf3 on slot A is SROS VM.

Volume in drive cf3 on slot A is formatted as FAT32

Directory of cf3:\recursive1

09/23/2021 02:42p ./
09/23/2021 02:42p ../
09/23/2021 02:42p 7 file1.txt
09/23/2021 02:42p recursive2/
1 File(s) 7 bytes.
3 Dir(s) 612298752 bytes free.

```

2.3.5 Moving files

Use the **move** command to move a file or directory from one location to another.

The **move** command supports wildcards, recursively moves files and directories, and overwrites existing content without prompting for confirmation.

Example: Moving files and directories

```

A:node-2>file cf3:\ # dir

Volume in drive cf3 on slot A is TIMOS_VM_CF.

```

```

Volume in drive cf3 on slot A is formatted as FAT32

Directory of cf3:\

09/28/2021  11:42a                933 B0F.CFG
09/28/2021  11:42a                166 NVRAM.DAT
09/29/2021  03:51p            12831 bootlog.txt
09/29/2021  03:46p            12828 bootlog_prev.txt
04/16/2014  10:15a                 0 config.cfg
09/29/2021  03:51p             318 nvsys.info
04/16/2014  10:15a      <DIR>      syslinux/
09/29/2021  03:55p            12831 test.txt
09/29/2021  03:54p      <DIR>      test_dir1/
04/16/2014  10:15a      <DIR>      timos/
                                7 File(s)          39907 bytes.
                                3 Dir(s)          14452736 bytes free.

A:node-2>file cf3:\ # move test.txt /test_dir1/test_dir2/test_dir3
Moving file cf3:\test.txt ... OK
cf3:\test.txt

A:node-2>file cf3:\ # dir

Volume in drive cf3 on slot A is TIMOS_VM_CF.

Volume in drive cf3 on slot A is formatted as FAT32

Directory of cf3:\

09/28/2021  11:42a                933 B0F.CFG
09/28/2021  11:42a                166 NVRAM.DAT
09/29/2021  03:51p            12831 bootlog.txt
09/29/2021  03:46p            12828 bootlog_prev.txt
04/16/2014  10:15a                 0 config.cfg
09/29/2021  03:51p             318 nvsys.info
04/16/2014  10:15a      <DIR>      syslinux/
09/29/2021  03:54p      <DIR>      test_dir1/
04/16/2014  10:15a      <DIR>      timos/
                                6 File(s)          27076 bytes.
                                3 Dir(s)          14452736 bytes free.

A:node-2>file cf3:\ # dir test_dir1/test_dir2/test_dir3

Volume in drive cf3 on slot A is TIMOS_VM_CF.

Volume in drive cf3 on slot A is formatted as FAT32

Directory of cf3:\test_dir1\test_dir2\test_dir3

09/29/2021  03:54p      <DIR>      ./
09/29/2021  03:54p      <DIR>      ../
09/29/2021  03:55p            12831 test.txt
                                1 File(s)          12831 bytes.
                                2 Dir(s)          14452736 bytes free.

```

Example: Recursive directory move

```

A:node-2>file cf3:\ # dir

Volume in drive cf3 on slot A is SROS VM.

Volume in drive cf3 on slot A is formatted as FAT32

```

```
Directory of cf3:\

09/19/2021 06:11p 196 B0F.CFG
09/19/2021 06:11p 0 CONFIG.CFG
09/19/2021 06:11p 101 NVRAM.DAT
09/19/2021 07:22p  SUPPORT/
09/19/2021 06:11p  SYSLINUX/
09/19/2021 06:11p  TIMOS/
09/23/2021 09:03a 27459 bootlog.txt
09/20/2021 09:56a 27326 bootlog_prev.txt
09/23/2021 01:21p 319 nvsys.info
09/23/2021 02:42p recursive1/
09/23/2021 08:22a  ssh/
7 File(s) 55402 bytes.
7 Dir(s) 612311040 bytes free.

A:node-2>file cf3:\ # dir recursive1

Volume in drive cf3 on slot A is SR0S VM.

Volume in drive cf3 on slot A is formatted as FAT32

Directory of cf3:\recursive1

09/23/2021 02:42p ./
09/23/2021 02:42p ../
09/23/2021 02:42p 7 file1.txt
09/23/2021 02:42p recursive2/
1 File(s) 7 bytes.
3 Dir(s) 612311040 bytes free.

A:node-2>file cf3:\ # move recursive1 recursive4
Moving file cf3:\recursive1 ... OK
cf3:\recursive1
A:sros>file cf3:\ # dir

Volume in drive cf3 on slot A is SR0S VM.

Volume in drive cf3 on slot A is formatted as FAT32

Directory of cf3:\

09/19/2021 06:11p 196 B0F.CFG
09/19/2021 06:11p 0 CONFIG.CFG
09/19/2021 06:11p 101 NVRAM.DAT
09/19/2021 07:22p  SUPPORT/
09/19/2021 06:11p  SYSLINUX/
09/19/2021 06:11p  TIMOS/
09/23/2021 09:03a 27459 bootlog.txt
09/20/2021 09:56a 27326 bootlog_prev.txt
09/23/2021 01:21p 319 nvsys.info
09/23/2021 02:42p recursive4/
09/23/2021 08:22a  ssh/
7 File(s) 55402 bytes.
7 Dir(s) 612311040 bytes free.

A:node-2>file cf3:\ # dir recursive4

Volume in drive cf3 on slot A is SR0S VM.

Volume in drive cf3 on slot A is formatted as FAT32

Directory of cf3:\recursive4
```



```
09/23/2021 02:42p ./
09/23/2021 02:42p ../
09/23/2021 02:42p 7 file1.txt
09/23/2021 02:42p recursive2/
1 File(s) 7 bytes.
3 Dir(s) 612311040 bytes free.
```

2.3.6 Deleting files and removing directories

Use the **delete** and **rd** commands to delete files and remove directories. Directories can be removed even if they contain files or subdirectories. To remove a directory that contains files and subdirectories, use the **rd /s** command. When files or directories are deleted, they cannot be recovered. The **force** option deletes the file or directory without prompting the user to confirm.

Example: Delete files and remove directories

```
A:node2>file cfl:\test1\ # delete test.cfg
A:node-2>file cfl:\test1\ # delete abc.cfg
A:node-2>file cfl:\test1\test2\ # cd test3
A:node-2>file cfl:\test1\test2\test3\ # cd ..
A:node-2>file cfl:\test1\test2\ # rd test3
A:node-2>file cfl:\test1\test2\ # cd ..
A:node-2>file cfl:\test1\ # rd test2
A:node-2>file cfl:\test1\ # cd ..
A:node-2>file cfl:\ # rd test1
A:node-2>file cfl:\ #
```

2.3.7 Unzipping files

Use the **unzip** command to expand the contents of a ZIP file to the local file system. Any file that is zipped using the store, deflate, or zip64 compression methods can be unzipped. An example is the SR OS software image available from the Nokia customer support portal.

The source ZIP file can be located locally on the installed solid-state storage device, or remotely on an FTP or TFTP server.

The **create-destination** keyword ensures that any non-existent directory structure that is explicitly entered as the destination file URL is created as part of the unzip operation.



Note:

- The destination for the unzipped files and directories must be a locally installed solid-state storage device in the active CPM.
- ZIP filenames, or the filenames of any contained files, must not include special characters.

Example: Unzip command

```
A:node-2# file unzip demo.zip cf3:/mynewfolder/mynewsfolder create-destination force
Verifying cf3:\demo.zip .. ... OK
Unzipping cf3:\demo.zip to cf3:\mynewfolder\mynewsfolder\ .. .Processing demodir/
Processing demodir/myfile1.txt
Processing demodir/myfile2.txt
Processing demodir/demosubdir/
Processing demodir/demosubdir/myfile3.txt
```

```
Writing...OK
```

2.3.8 Repairing the file system

Use the **repair** command to check a compact flash device for errors and repair any errors found.

Example: Command syntax

```
[file "cf3:\"]
A:admin@node-2# repair cflash-id cf3:
Checking drive cf3: on slot A for errors...
Drive cf3: on slot A is OK.

[file "cf3:\"]
A:admin@node-2# repair
Checking drive cf3: on slot A for errors...
Drive cf3: on slot A is OK.
```

2.3.9 Displaying file checksums

Use the **checksum** command to display file checksums.

Use the **version** command to check the version of an SR OS .tim image file.

Example: Checking the version of an SR OS .tim image file

```
A:node-2>file cf3:\ # version cpm.tim
TiMOS-C-20.10.R1
Wed Nov 4 09:18:17 PST 2020 by builder in /builds/c/2010B/R1/panos/main/sros

A:node-2>file cf3:\ # version cpm.tim check
TiMOS-C-20.10.R1
Wed Nov 4 09:18:17 PST 2020 by builder in /builds/c/2010B/R1/panos/main/sros
Checking file ... OK
```

Use the **checksum** command to display checksums.

Example: Output of the checksum operation

The following example shows the output of the checksum operation to compute and display a checksum based on the MD5 and SHA256 algorithms for the `config.cfg` file on `cf3`.

```
A:node-2# file checksum md5 cf3:config.cfg
Checking file cf3:config.cfg ...
c65699dc05e6e35a2172eaac80485aa2

A:node-2# file checksum sha256 cf3:config.cfg
Checking file cf3:config.cfg ...
a1a813a696be04906f9faf1df9db0f90a990ff51cb3383099ade21241203bc1c
```

2.4 File management tasks in the MD-CLI

The following sections describe basic system tasks that can be performed in the MD-CLI.

For more information about the supported MD-CLI commands, see the *7705 SAR Gen 2 MD-CLI Command Reference Guide*.

When a file system operation is performed that can potentially remove or overwrite a file system entry, a prompt appears to confirm the action. The **force** keyword performs the operation without displaying the confirmation prompt.

All the commands can operate on the local file system. The following table lists which commands also support remote file operations.

Table 14: File command local and remote file system support in the MD-CLI

Command	local-url	ftp-url	tftp-url	http-url	https-url
change-directory	✓	✓			
checksum	✓	✓	✓		
copy	✓	✓	source only	✓	✓
copy (recursive)	✓	✓			
list	✓	✓			
make-directory	✓	✓			
move	✓	✓		✓	✓
move (recursive)	✓	✓			
permission	✓				
remove	✓	✓		✓	✓
remove-directory	✓	✓			
show	✓	✓	✓	✓	✓
unzip	✓	✓	source only		
version	✓	✓	✓		

2.4.1 Displaying directory and file information

Use the **list** command to list the files on a file system, with an option to indicate the list order based on the date, name, or size of the files. The **show** command displays the contents of a specified file or multiple files. The **version** command displays the version of an SR OS .tim image file.

Example: Display directory and file information

```
[file "cf3:\"]
A:admin@node-2# list

Volume in drive cf3 on slot A is .

Volume in drive cf3 on slot A is formatted as FAT32
```

```

Directory of cf3:\

09/01/2020  11:27p      <DIR>          .ssh/
01/01/1980  12:00a                170 NVRAM.DAT
01/01/1980  12:00a                679 bof.cfg
09/01/2020  11:27p                319 nvsys.info
09/01/2020  11:27p                1 restcntr.txt
09/02/2020  04:32p      <DIR>          tstdir/
                                4 File(s)          1169 bytes.
                                2 Dir(s)           0 bytes free.

[file "cf3:\"]
A:admin@node-2# list size

Volume in drive cf3 on slot A is .

Volume in drive cf3 on slot A is formatted as FAT32

Directory of cf3:\

09/01/2020  11:27p      <DIR>          .ssh/
09/02/2020  04:32p      <DIR>          tstdir/
09/01/2020  11:27p                1 restcntr.txt
01/01/1980  12:00a                170 NVRAM.DAT
09/01/2020  11:27p                319 nvsys.info
01/01/1980  12:00a                679 bof.cfg
                                4 File(s)          1169 bytes.
                                2 Dir(s)           0 bytes free.

[file "cf3:\"]
A:admin@node-3# show md-config.cfg
File: md-config.cfg
-----
configure {
    card 1 {
        mda 1 {
        }
    }
    log {
        filter 1001 {
            entry 10 {
                description "Collect only events of major severity or higher"
                action forward
                match {
                    severity {
                        gte major
                    }
                }
            }
        }
    }
    log-id 99 {
        description "Default System Log"
        source {
            main true
        }
    }
}
Press Q to quit, Enter to print next line or any other key to print next page.

```

2.4.2 Modifying file attributes

The system administrator can change the attribute of a local file or files in a directory.

Enter the **permission** command with no options to display the contents of the directory and the file attributes.

A single local file can be specified or the wildcard character (*) can be used to indicate multiple files. If no URL is specified, the command applies to all files in the directory.

A file with an "R" preceding the filename indicates the file is read-only; otherwise, the file is read-write.

Example: Modify file attributes

```
[file "cf3:\"]
A:admin@node-2# permission
      cf3:\NVRAM.DAT
      cf3:\bof.cfg
      cf3:\nvsys.info
      cf3:\restcntr.txt
      cf3:\.ssh
      cf3:\my.txt

[file "cf3:\"]
A:admin@node-2# permission read-only my.txt

[file "cf3:\"]
A:admin@node-2# permission
      cf3:\NVRAM.DAT
      cf3:\bof.cfg
      cf3:\nvsys.info
      cf3:\restcntr.txt
      cf3:\.ssh
      cf3:\my.txt
R
```

```
[file "cf3:\"]
A:admin@node-2# permission read-only

[file "cf3:\"]
A:admin@node-2# permission
R      cf3:\NVRAM.DAT
R      cf3:\bof.cfg
R      cf3:\nvsys.info
R      cf3:\restcntr.txt
R      cf3:\.ssh
R      cf3:\my.txt
```

2.4.3 Creating and navigating directories

New directories can be created in the local file system, one level at a time.

Use the **make-directory** command to create a new directory.

The **change-directory** command navigates to different directories.

Example: Create and navigate directories

```
[file "cf3:\"]
A:admin@node-2# make-directory test1

[file "cf3:\"]
A:admin@node-2# change-directory test1

[file "cf3:\test1"]
```

```

A:admin@node-2# make-directory test2

[file "cf3:\test1"]
A:admin@node-2# change-directory test2

[file "cf3:\test1\test2"]
A:admin@node-2# make-directory test3

[file "cf3:\test1\test2"]
A:admin@node-2# change-directory test3

[file "cf3:\test1\test2\test3"]
A:admin@node-2# change-directory ..

[file "cf3:\test1\test2"]
A:admin@node-2#

```

2.4.4 Copying files

Use the **copy** command to copy files to or from a flash device, or an FTP, TFTP, or SSH server. The **copy** command supports wildcards.

Use the **recursive** option to recursively copy files and directories. If files or directories already exist, the user is prompted to overwrite them. Use the **force** option to automatically overwrite the existing files or directories, without being prompted for confirmation.

The following example shows how to copy the `config.cfg` file to the `test_dir1` directory.

Example: Local file copy

```

[/]
A:admin@node-2# file list test_dir1

Volume in drive cf3 on slot A is TIMOS_VM_CF.

Volume in drive cf3 on slot A is formatted as FAT32

Directory of cf3:\test_dir1

09/29/2021  05:07p      <DIR>          ./
09/29/2021  05:07p      <DIR>          ../
                0 File(s)                  0 bytes.
                2 Dir(s)                14458880 bytes free.

[/]
A:admin@node-2# file list

Volume in drive cf3 on slot A is TIMOS_VM_CF.

Volume in drive cf3 on slot A is formatted as FAT32

Directory of cf3:\

09/28/2021  11:43a           931 B0F.CFG
09/28/2021  11:43a           165 NVRAM.DAT
09/28/2021  04:34p          11319 bootlog.txt
09/28/2021  03:50p          11259 bootlog_prev.txt
09/29/2021  05:08p          11319 config.cfg
09/28/2021  04:34p           319 nvsys.info
04/16/2014  10:15a      <DIR>          syslinux/

```

```

09/29/2021 05:08p <DIR> test_dir1/
04/16/2014 10:15a <DIR> timos/
              7 File(s)          35312 bytes.
              3 Dir(s)          14458880 bytes free.

[/]
A:admin@node-2# file copy config.cfg test_dir1
Copying file cf3:\config.cfg ... OK
1 file copied.

[/]
A:admin@node-2# file list test_dir1

Volume in drive cf3 on slot A is TIMOS_VM_CF.

Volume in drive cf3 on slot A is formatted as FAT32

Directory of cf3:\test_dir1

09/29/2021 05:07p <DIR> ./
09/29/2021 05:07p <DIR> ../
09/29/2021 05:09p      11319 config.cfg
              1 File(s)          11319 bytes.
              2 Dir(s)          14447104 bytes free.

```

The following example shows how to copy the config.cfg file to a file on an SSH server.

Example: File copy to SSH server

```

[/]
A:admin@node-2# file copy cf3:config.cfg scp://user@10.231.216.68:~/
user@10.231.216.68's password:
config.cfg                                100% 625      0.6KB/s   00:00

```

The following example shows a recursive move of the recursive4 directory to recursive5.

Example: Recursive directory move

```

[/]
A:admin@node-2# file list

Volume in drive cf3 on slot A is SROS VM.

Volume in drive cf3 on slot A is formatted as FAT32

Directory of cf3:\

09/19/2021 06:11p 196 B0F.CFG
09/19/2021 06:11p 0 CONFIG.CFG
09/19/2021 07:22p SUPPORT/
09/23/2021 09:03a 27459 bootlog.txt
09/20/2021 09:56a 27326 bootlog_prev.txt
09/23/2021 01:21p 319 nvsys.info
09/23/2021 02:42p recursive4/
09/23/2021 08:22a ssh/
7 File(s) 55402 bytes.
7 Dir(s) 612311040 bytes free.

[/]
A:admin@node-2# file copy recursive4 recursive5 recursive
Copying directory cf3:\recursive4
Copying directory cf3:\recursive4\recursive2

```

```
Copying file cf3:\recursive4\recursive2\file2.txt ... OK
Copying file cf3:\recursive4\recursive2\file3.txt ... OK
Copying file cf3:\recursive4\file1.txt ... OK
2 dir(s) and 3 file(s) copied.
```

2.4.5 Moving files

Files or directories can be moved from one location to another. The **move** command recursively moves files and directories, and overwrites existing content without prompting for confirmation. The **move** command supports wildcards.

The following example moves the `md-config.cfg` file to the `test_dir1` directory.

Example: Moving files and directories

```
[/file "cf3:\"]
A:admin@node-2# list test_dir1

Volume in drive cf3 on slot A is TIMOS_VM_CF.

Volume in drive cf3 on slot A is formatted as FAT32

Directory of cf3:\test_dir1

09/29/2021  05:07p      <DIR>          ./
09/29/2021  05:07p      <DIR>          ../
              0 File(s)                  0 bytes.
              2 Dir(s)                 14458880 bytes free.

[/file "cf3:\"]
A:admin@node-2# list

Volume in drive cf3 on slot A is TIMOS_VM_CF.

Volume in drive cf3 on slot A is formatted as FAT32

Directory of cf3:\

09/28/2021  11:43a              931 B0F.CFG
09/28/2021  11:43a              165 NVRAM.DAT
09/28/2021  04:34p            11319 bootlog.txt
09/28/2021  03:50p            11259 bootlog_prev.txt
09/29/2021  05:08p            11319 md-config.cfg
09/28/2021  04:34p              319 nvsys.info
04/16/2014  10:15a      <DIR>          syslinux/
09/29/2021  05:09p      <DIR>          test_dir1/
04/16/2014  10:15a      <DIR>          timos/
              7 File(s)                 35312 bytes.
              3 Dir(s)                 14458880 bytes free.

[/file "cf3:\"]
A:admin@node-2# move md-config.cfg test_dir1
Moving file cf3:\md-config.cfg ... OK
cf3:\md-config.cfg

[/file "cf3:\"]
A:admin@node-2# list test_dir1

Volume in drive cf3 on slot A is TIMOS_VM_CF.
```



```

Volume in drive cf3 on slot A is formatted as FAT32

Directory of cf3:\test_dir1

09/29/2021  05:07p      <DIR>      ./
09/29/2021  05:07p      <DIR>      ../
09/29/2021  05:08p      11319 md-config.cfg      11319 bytes.
                1 File(s)
                2 Dir(s)      14458880 bytes free.

[/file "cf3:\"]
A:admin@node-2# list

Volume in drive cf3 on slot A is TIMOS_VM_CF.

Volume in drive cf3 on slot A is formatted as FAT32

Directory of cf3:\

09/28/2021  11:43a      931 B0F.CFG
09/28/2021  11:43a      165 NVRAM.DAT
09/28/2021  04:34p     11319 bootlog.txt
09/28/2021  03:50p     11259 bootlog_prev.txt
09/28/2021  04:34p      319 nvsys.info
04/16/2014  10:15a      <DIR>      syslinux/
10/01/2021  04:17p      <DIR>      test_dir1/
04/16/2021  10:15a      <DIR>      timos/
                6 File(s)      23993 bytes.
                3 Dir(s)      14458880 bytes free.

```

The following example shows a recursive move of the recursive1 directory to recursive3.

Example: Recursive directory move

```

[/file "cf3:\"]
A:admin@node-2# list

Volume in drive cf3 on slot A is SROS VM.

Volume in drive cf3 on slot A is formatted as FAT32

Directory of cf3:\

09/19/2021  06:11p  196 B0F.CFG
09/19/2021  06:11p   0 CONFIG.CFG
09/19/2021  07:22p  SUPPORT/
09/20/2021  09:56a 27326 bootlog.txt
09/20/2021  02:55p  319 nvsys.info
09/21/2021  07:19p recursive1/
09/20/2021  09:55a ssh/
 6 File(s) 27943 bytes.
 6 Dir(s) 612347904 bytes free.

[/file "cf3:\"]
A:admin@node-2# list recursive1

Volume in drive cf3 on slot A is SROS VM.

Volume in drive cf3 on slot A is formatted as FAT32

Directory of cf3:\recursive1

```

```
09/21/2021 07:18p ./
09/21/2021 07:18p ../
09/21/2021 07:19p 7 file1.txt
09/21/2021 07:19p recursive2/
1 File(s) 7 bytes.
3 Dir(s) 612347904 bytes free.

[/file "cf3:\"]
A:admin@node-2# list recursive1/recursive2

Volume in drive cf3 on slot A is SROS VM.

Volume in drive cf3 on slot A is formatted as FAT32

Directory of cf3:\recursive1\recursive2

09/21/2021 07:19p ./
09/21/2021 07:19p ../
09/21/2021 07:19p 7 file2.txt
09/21/2021 07:19p 7 file3.txt
2 File(s) 14 bytes.
2 Dir(s) 612347904 bytes free.

[/file "cf3:\"]
A:admin@sros# move recursive1 recursive3
Moving file cf3:\recursive1 ... OK
cf3:\recursive1

[/file "cf3:\"]
A:admin@node-2# list

Volume in drive cf3 on slot A is SROS VM.

Volume in drive cf3 on slot A is formatted as FAT32

Directory of cf3:\

09/19/2021 06:11p 196 B0F.CFG
09/19/2021 06:11p 0 CONFIG.CFG
09/19/2021 06:11p 101 NVRAM.DAT
09/19/2021 07:22p SUPPORT/
09/19/2021 06:11p SYSLINUX/
09/19/2021 06:11p TIMOS/
09/20/2021 09:56a 27326 bootlog.txt
09/20/2021 02:55p 319 nvsys.info
09/21/2021 07:19p recursive3/
09/20/2021 09:55a ssh/
6 File(s) 27943 bytes.
6 Dir(s) 612347904 bytes free.

[/file "cf3:\"]
A:admin@node-2# list recursive3

Volume in drive cf3 on slot A is SROS VM.

Volume in drive cf3 on slot A is formatted as FAT32

Directory of cf3:\recursive3

09/21/2021 07:18p ./
09/21/2021 07:18p ../
09/21/2021 07:19p 7 file1.txt
09/21/2021 07:19p recursive2/
1 File(s) 7 bytes.
```

```
3 Dir(s) 612347904 bytes free.
```

2.4.6 Deleting files and removing directories

Use the **remove** and **remove-directory** commands to delete files and remove directories. Directories can be removed even if they contain files or subdirectories.

Example: Removing directories

```
[file "cf3:\test1\test2"]
A:admin@node-2# list

Volume in drive cf3 on slot A is TIMOS_VM_CF.

Volume in drive cf3 on slot A is formatted as FAT32

Directory of cf3:\test1\test2

09/01/2020  08:13p      <DIR>          ./
09/01/2020  08:13p      <DIR>          ../
09/04/2020  06:36p                874 bof.cfg
04/28/2020  03:15p            11401 md-config.cfg
09/04/2020  06:43p      <DIR>          test3/
                2 File(s)                12275 bytes.
                3 Dir(s)                10641920 bytes free.

[file "cf3:\test1\test2"]
A:admin@node-2# list test3

Volume in drive cf3 on slot A is TIMOS_VM_CF.

Volume in drive cf3 on slot A is formatted as FAT32

Directory of cf3:\test1\test2\test3

09/01/2020  08:13p      <DIR>          ./
09/01/2020  08:13p      <DIR>          ../
09/04/2020  06:43p            11788 conf3.cfg
09/04/2020  04:24p            6645 mybof.cfg
                2 File(s)                18433 bytes.
                2 Dir(s)                10641920 bytes free.

[file "cf3:\test1\test2"]
A:admin@node-2# remove-directory test3 ?

remove-directory
force              - Force removal without prompting
recursive          - Remove directory and its content recursively

[file "cf3:\test1\test2"]
A:admin@node-2# remove-directory test3 recursive
Deleting all subdirectories and files in specified directory. y/n ?y
Deleting file cf3:\test1\test2\test3\mybof.cfg ... OK
Deleting file cf3:\test1\test2\test3\conf3.cfg ... OK
Deleting directory cf3:\test1\test2\test3 ... OK

[file "cf3:\test1\test2"]
A:admin@node-2# list

Volume in drive cf3 on slot A is TIMOS_VM_CF.
```

```

Volume in drive cf3 on slot A is formatted as FAT32

Directory of cf3:\test1\test2

09/01/2020  08:13p      <DIR>          ./
09/01/2020  08:13p      <DIR>          ../
09/04/2020  06:36p                   874  bof.cfg
04/28/2020  03:15p            11401 md-config.cfg
                2 File(s)                  12275 bytes.
                2 Dir(s)                 10661376 bytes free.

```

2.4.7 Unzipping files

Use the **unzip** command to expand the contents of a ZIP file to the local file system. Any file zipped using the store, deflate, or zip64 compression methods can be unzipped. An example is the SR OS software image available from the Nokia customer support portal.

The source ZIP file location can be a locally installed solid-state storage device or a remote FTP or TFTP server.

The **create-destination** keyword ensures that any non-existent directory structure that is explicitly entered as the destination file URL is created as part of the unzip operation. This parameter is required to create new directories.



Note:

- The destination for the unzipped files and directories must be a locally installed solid-state storage device in the active CPM.
- ZIP filenames, or the filenames of any contained files, must not include special characters.

Example: Unzipping a file

```

[file "cf3:\"]
A:admin@node-2# unzip demo.zip cf3:/mynewfolder/mynewsfolder create-
destination force
Verifying cf3:\demo.zip .. ... OK
Unzipping cf3:\demo.zip to cf3:/mynewfolder/mynewsfolder\ .. .Processing demodir/
Processing demodir/myfile1.txt
Processing demodir/myfile2.txt
Processing demodir/demosubdir/
Processing demodir/demosubdir/myfile3.txt
Writing...OK

```

2.4.8 Repairing the file system

Use the **repair** command to check a compact flash device for errors and repair any errors found.

Example: Command syntax

```

[file "cf3:\"]
A:admin@node-2# repair cflash-id cf3:
Checking drive cf3: on slot A for errors...
Drive cf3: on slot A is OK.

[file "cf3:\"]

```

```
A:admin@node-2# repair
Checking drive cf3: on slot A for errors...
Drive cf3: on slot A is OK.
```

2.4.9 Displaying file checksums

Use the **checksum** command to display file checksums.

Example: Check a .tim image file checksum

The following example shows the output of the operation to check an SR OS .tim image file checksum.

```
[file "cf3:\"]
A:admin@node-2# checksum image cpm.tim
TiMOS-C-20.2.R1
Sat Feb 29 10:39:32 PST 2020 by builder in /builds/c/202B/R1/panos/main/sros
Checking file ... OK
```

Example: Output of the checksum operation

The following example shows the output of the checksum operation to compute and display a checksum based on the MD5 and SHA256 algorithms.

```
[/]
A:admin@node-2# file checksum md5 cf3:config.cfg
Checking file cf3:config.cfg
c65699dc05e6e35a2172eaac80485aa2

[/]
A:admin@node-2# file checksum sha256 cf3:config.cfg
Checking file cf3:config.cfg
a1a813a696be04906f9faf1df9db0f90a990ff51cb3383099ade21241203bc1c
```

3 System initialization and boot options

This section describes the system initialization and boot option process.

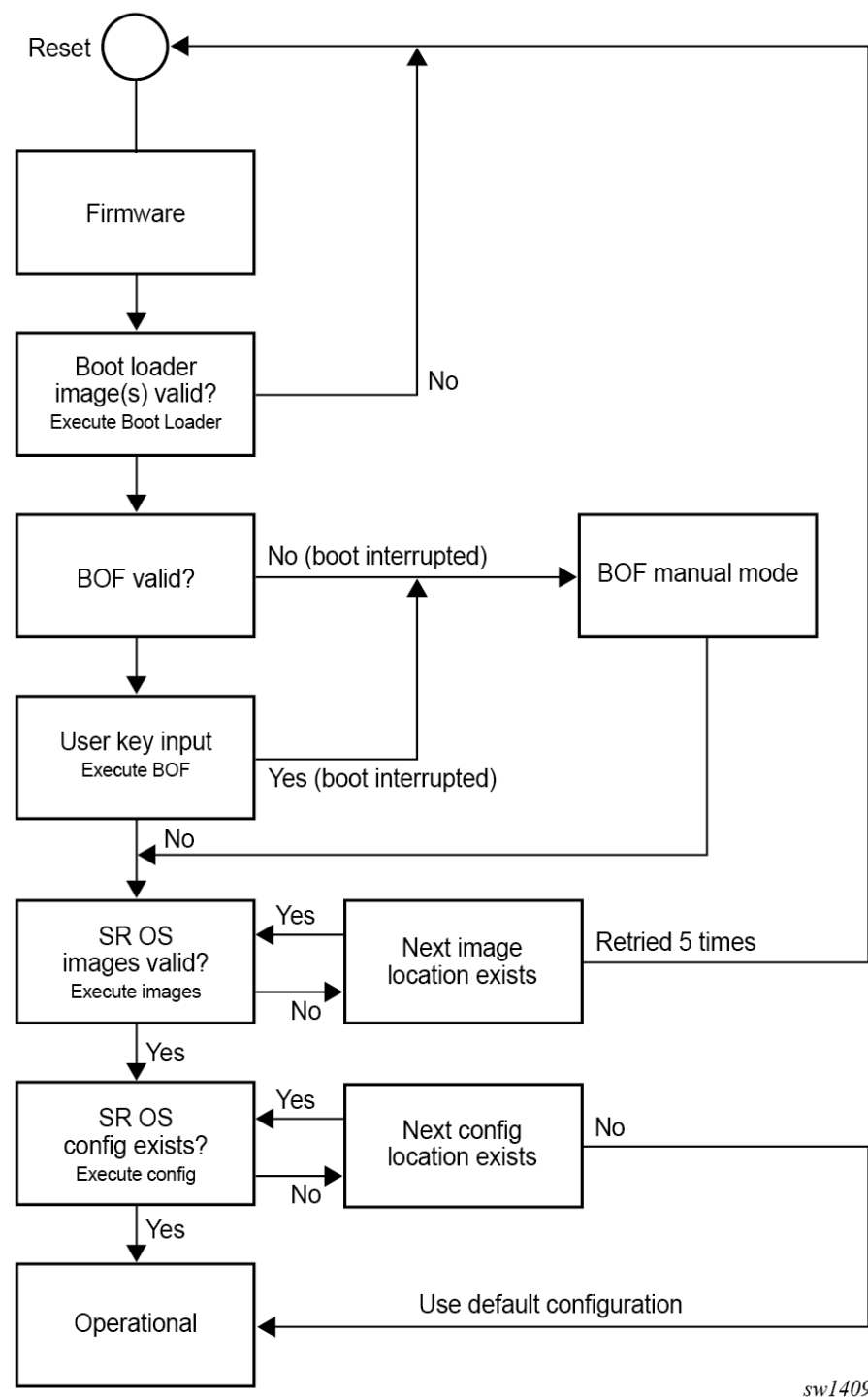
3.1 Boot process

The router startup process begins after a reset or power cycle with the firmware initializing the hardware before executing the Boot Loader images. The Boot Loader then executes the Boot Options File (BOF) to load the SR OS software image and configuration. The BOF file contains system initialization commands including the software image and configuration locations.

SR OS Boot Loader, software images, and configuration files are stored in storage media cards referred to as CF in the system. See [Storage devices](#) for more information about the type of storage supported for each platform.

The following diagram shows the system boot process from the firmware up to the SR OS image and configuration file.

Figure 1: Boot process



sw1409

3.2 Boot Loader

The Boot Loader executes the initialization parameters from the BOF to load the software images and configuration file.

The Boot Loader phase can be manually interrupted even if the BOF is present by pressing any key on the console connected to the console port. This is done by typing **sros** and pressing the **Enter** key within 30 seconds to enter the BOF Manual Mode. This mode allows the configuration of the BOF system initialization commands manually and overwrites the existing BOF file if present.

Different Boot Loader images are used depending on the CPM control module:

- 7705 SAR-1
 - `bootaa64.efi` is the original Boot Loader image located in `/EFI/B00T`
 - `bootaa64.efi` executes the other images in `/EFI/B00T/aarch64` before executing the BOF file and loading SR OS TiMOS software images and configuration

3.3 Boot Options File

The BOF file (`bof.cfg`) must be located at the root of the CF3 card directory and contains various system initialization commands including:

- management Ethernet port (speed, duplex, IP address, static routes)
- console port speed
- software image locations
- configuration file locations
- BOF and configuration file encryption settings
- BOF password
- system profile
- wait time
- Zero Touch Provisioning
- licenses
- persistency

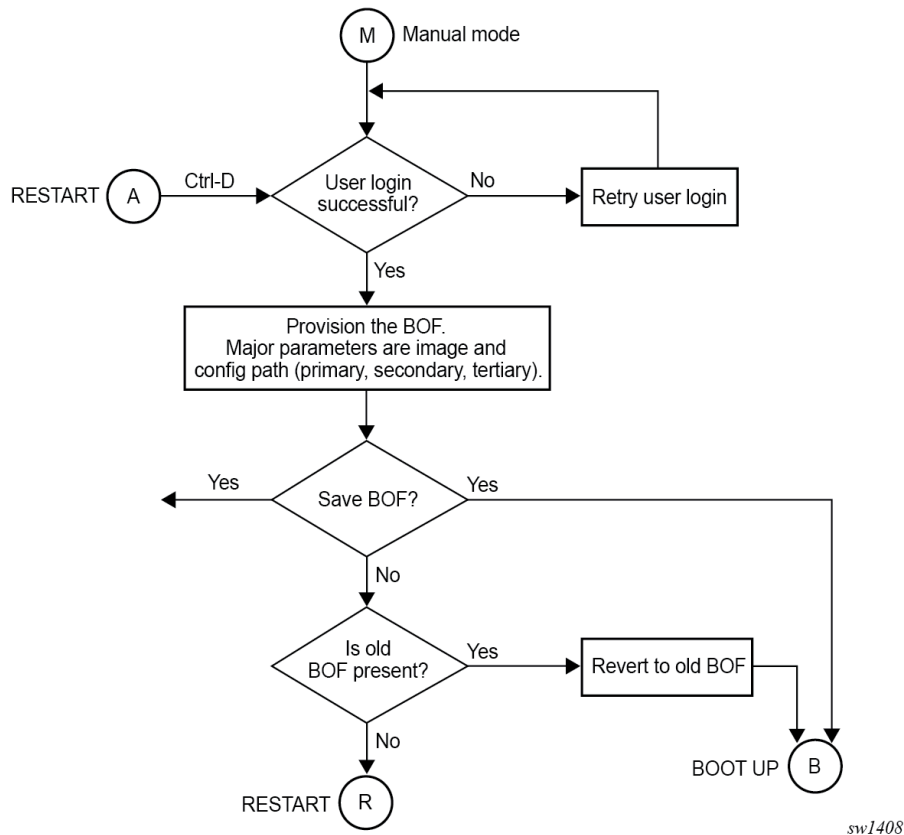
3.3.1 BOF manual mode

The system enters the BOF manual mode if the BOF is not present on `cf3:\` or if the user interrupts the Boot Loader phase and requires access to the console port for configuration.

After the manual BOF configuration is completed and saved, a `bof.cfg` file with the new configured command options is created on `cf3:\` and used for subsequent reboots. The Boot Loader image then processes the new BOF command options to boot the system.

This process is described in the following diagram.

Figure 2: BOF manual mode



3.4 Software and configuration

The software image and configuration file location are configured in the BOF.

Up to three locations, local or remote, can be configured for the software image and configuration file defined as primary image, secondary image, tertiary image, primary config, secondary config and tertiary config in the BOF.

Before loading the configuration, the software first attempts to read the license file if one has been included in the BOF. If a license file is found, it is activated. If there are any issues with the activation, a log event is raised, and the startup processing continues with the reading of the configuration file.

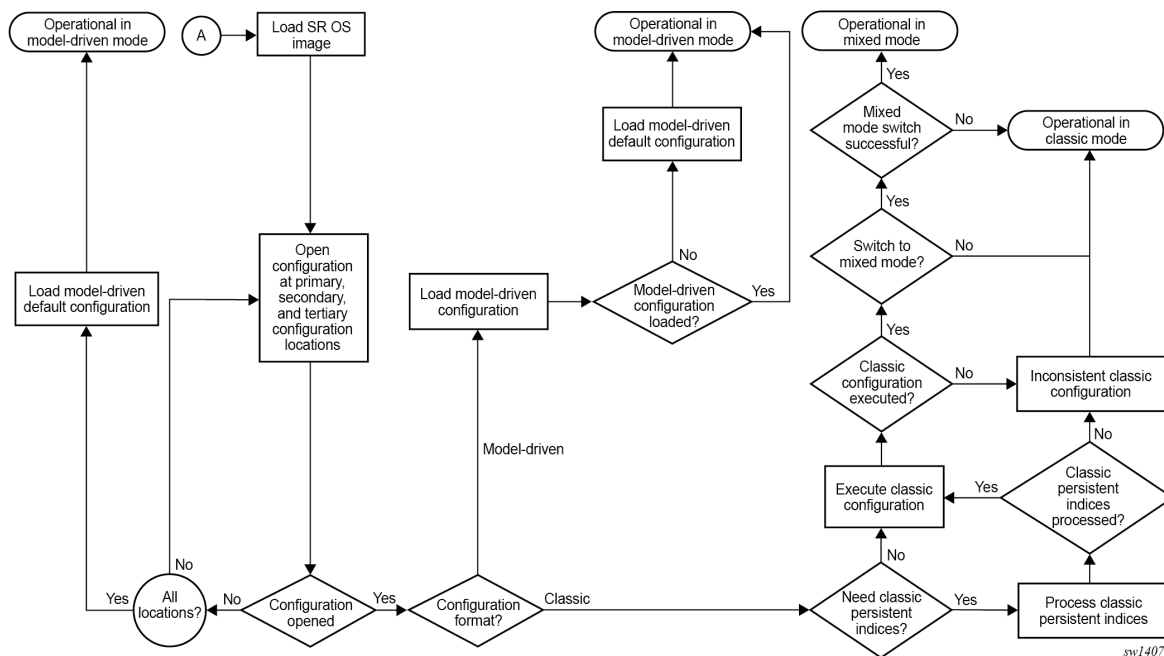
The following usage guidelines apply:

- The primary, secondary, and tertiary image locations must have the same version of software. If the secondary or tertiary image is configured with an older software image, this may result in a failure to load the configuration file as the file may contain commands only applicable to the more recent release.
- Similarly, the secondary and tertiary configuration files, if used, should be saved with the same version of software as the software executed on boot as it can result in a failure to load the configuration file otherwise.

- In the model-driven configuration mode, with incremental saved configuration files enabled, the primary configuration location supports complete and incremental saved configuration files. The secondary and tertiary configuration locations support complete saved configuration files. The user must ensure that complete saved configuration files are stored at both locations.

The following diagram provides additional details on the boot process differences between classic and MD-CLI configuration file processing.

Figure 3: Load SR OS configuration



3.4.1 Management interface configuration modes

The system can operate in different management interface configuration modes, which affect the CLI and network management protocols that can be used to configure the system. When the system boots and loads the configuration file, the configuration mode is set as follows:

- The default configuration mode is **model-driven** and a model-driven configuration file format is loaded. The value of **configure system management-interface configuration-mode** in the configuration file must not be **classic** or **mixed**.
- If the configuration file has **exit all** as the first executable line, the configuration mode is set to **classic** and a classic configuration file format is loaded. Lines beginning with a number sign character (#) are ignored.
 - The configuration mode may be changed to **mixed** if the value of **configure system management-interface configuration-mode** is **mixed** in the configuration file.
 - The value of **configure system management-interface configuration-mode** in the configuration file must not be **model-driven**.

See "Management interface configuration modes" in the *7705 SAR Gen 2 System Management Guide* for more information.

3.5 Initial installation and software update

SR OS is preinstalled on 7705 SAR Gen 2 systems from Nokia. The user is not required to perform initial software installation.

The 7705 SAR Gen 2 boot files are released by Nokia in a cflash package. When upgrading or changing software, the boot files must be placed in the root of cf3:/. The BOF primary location must be configured to the location of the boot files.

3.6 USB recovery boot

Prerequisites

- USB drive installed in cf:2\ containing the cflash package from Nokia that includes the boot files for the desired SR OS software release. The cflash package must be extracted and placed in the root of the USB drive. The files are preconfigured by Nokia to use the correct boot settings and should not be modified for the recovery procedure.



WARNING: The USB recovery procedure erases the contents of the eMMC. Nokia recommends creating a backup of the eMMC contents, if possible, to prevent data loss.

About this task

The 7705 SAR Gen 2 supports a boot recovery procedure using the USB drive installed on cf2:\. This procedure should be used only if the eMMC on cf3:\ becomes corrupt and the system fails to boot.

Perform the following steps to recover the eMMC to a bootable state.

Procedure

Step 1. Power on the system.

Step 2. On the boot screen, press the Up arrow key to enter the Recovery menu when the screen displays the following message: Autoboot in # seconds.

Example

```
Booting /MemoryMapped(0xa0000000, 0x199000)

Boot rom version is v72
Nokia 7xxx Boot ROM. Copyright 2020-2025 Nokia
All rights reserved. All use is subject to applicable license agreements.
X-0.0.18017 on Sat Feb 1 02:15:28 UTC 2025 by builder

INFO: Board type 0x24 [xxxxx_r3]
Autoboot in 3 seconds
```

Step 3. In the Recovery Menu, enter 1 to select the Recover using USB(cf2) option.

Example

```
-----
Recovery Menu:
-----
1. Recover using USB(cf2)
2. Exit
```

```
Enter valid recovery drive option: 1
```

Step 4. Enter Y to confirm the recovery process.

Example

```
-----
Recovery Menu:
-----
1. Recover using USB(cf2)
2. Exit

Enter valid recovery drive option: 1

[WARNING]: eMMC(cf3) will be erased during recovery. Ensure to have a backup before
proceeding.
Continue to recover eMMC(cf3) using USB(cf2) - y/ [N] ? y
```

Expected outcome

Upon user confirmation in step 4, the system reformats the eMMC using the files from the USB drive, then initiates the boot process.

3.7 Storage card content

SR OS software downloaded from the Nokia support website includes boot and operating system images for all platforms. This section describes the required storage media card directory and filenames on a per-platform basis to clarify which files and directory apply to which platform.

On the 7705 SAR Gen 2, the primary copy of the SR OS software is located on the eMMC on cf3:/.

Configurations and executable images can be stored in any storage media supported by the platform while the boot loader images, and boot option file must be installed in the cf3:.

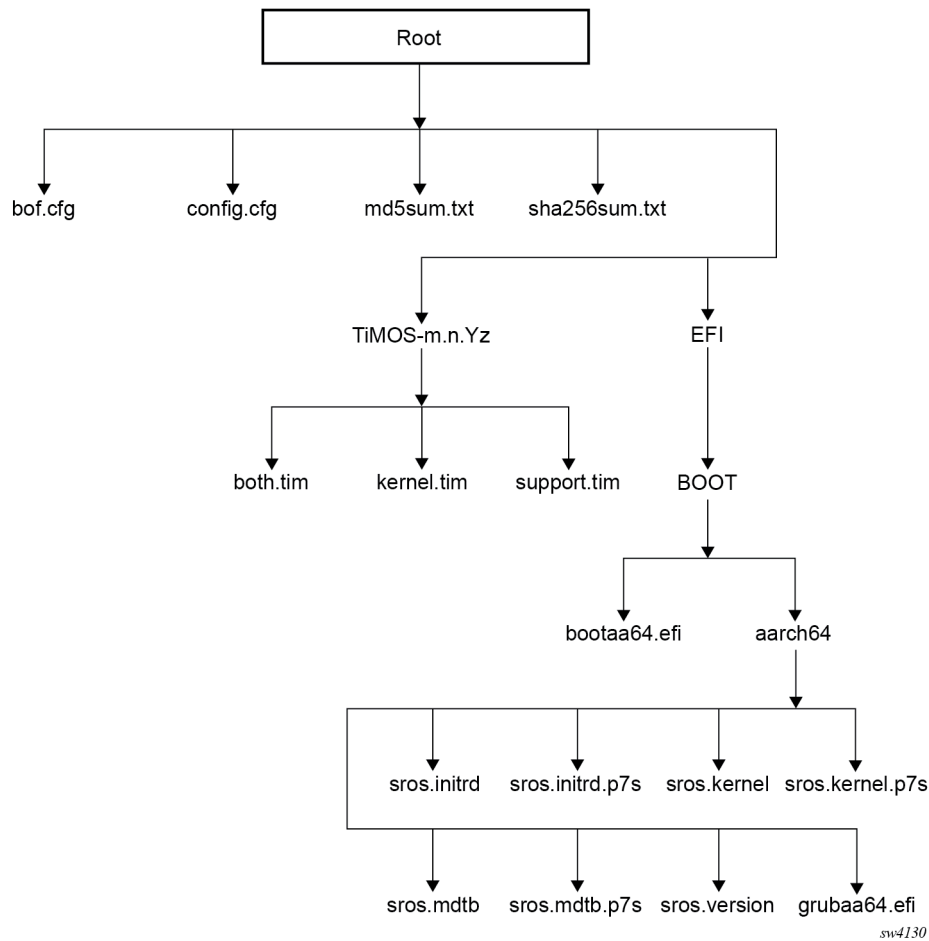
See the [Storage devices](#) section for the list of storage media names, locations, and support.

3.7.1 7705 SAR-1 storage card content

When installing a new storage media card into the system for the first time, ensure that the media card contains only the software files shipped by Nokia.

The following figure shows the required storage media card directory structure and filenames.

Figure 4: Files on storage card — 7705 SAR-1



The following files are present on the storage media card:

- `bof.cfg` – boot option file
- `config.cfg` – default configuration file
- `md5sum.txt` – MD5 checksum file
- `sha256sum.txt` – SHA256 checksum file
- `TiMOS-m.n.Yz`:
 - m – signifies a major release number
 - n – signifies a minor release number
 - Y: A signifies an alpha release
 - B – signifies a beta release
 - M – signifies a maintenance release
 - R – signifies a released software
 - z – signifies a version number

- `cpm.tim` – CPM and IOM image file
- `kernel.tim` – host operating system
- `support.tim` – required data for SR OS .tim files

The following files and folder structure under `/EFI` should only be included if the system is booted with a new storage media card installed for the first time:

- `EFI`:
 - `BOOT`:
 - `bootaa64.efi` – EFI file; Boot Loader
 - `aarch64`
 - `sros.mdtb` - device tree blob
 - `sros.initrd` – OS installer file; installer rootfs
 - `sros.kernel` – OS installer file; installer kernel
 - `sros.version` – OS installer file; installer version
 - `grubaa64.efi` – EFI file; GRUB boot loader
 - `sros.kernel.p7s` - OS installer file; installer kernel digital signature
 - `sros.initrd.p7s` - OS installer file; installer rootfs digital signature
 - `sros.mdtb.p7s` - device tree blob digital signature

3.7.2 ISA and ESA applications

The following images must also be present in the SR OS software `TiMOS-m-n-Yz` directory depending if ESA or Application Assurance are used:

- `hypervisors.tim`: hypervisor image for the ESA cards
- `isa-aa.tim`: Application Assurance software image

3.8 Persistent indexes in classic and mixed configuration mode

The BOF **persist** command option specifies whether the system should preserve system indexes when the configuration is saved. During a subsequent boot, the index file is read along with the configuration file. As a result, a number of system indexes are preserved between reboots, including the SNMP interface index, LSP IDs, path IDs, and so on. If persistence is not required and the configuration file is successfully processed, the system becomes operational. If **persist** is required, a matching `config.ndx` file must be successfully processed before the system can become operational. Configuration and index files must have the same filename prefix such as `config.cfg` and `config.ndx` and are created at the same time when an **admin save** command is executed. Note that the persistence option must be enabled to deploy a Network Management System (NMS) using SNMP. The default is off.



Note: System indexes in model-driven configuration mode are always persistent.

3.9 BOF and configuration file encryption

In cases where the platform is not installed in a physically secure location, the user can encrypt the BOF and the configuration file to halt or hinder interpretation of the file content.

By default, the BOF and configuration files are not encrypted. When encryption is enabled for either file and a change is saved, the original file is moved to *filename.1* and the encrypted file becomes the new *filename.cfg*.



Caution: The first time a file is encrypted and the original file is moved to *filename.1*, the *filename.1* file is unencrypted. Delete the unencrypted file to maintain node security.

When the BOF is encrypted on the Compact Flash, the BOF interactive menu can be used during node startup to access the file and modify BOF fields. To prevent unauthorized modification of the BOF using the BOF interactive menu, configure a password using the following command:

- **MD-CLI**

```
bof configuration password
```

- **classic CLI**

```
bof password
```

The BOF interactive menu is accessible only when the configured password is entered. If the correct password is not entered in 30 seconds, the node reboots.

See [Configuring BOF encryption](#) for information about configuring BOF encryption. See [Configuring the BOF interactive menu password](#) for information about configuring the BOF interactive menu password. See [Configuring configuration file encryption](#) for information about configuring configuration file encryption.

3.10 System profiles

System profiles provide flexibility when using line cards by supporting different system capabilities. The system profile is defined in the BOF and is used by the system when it is next rebooted. Contact your Nokia representative for system profile information.

The following system profiles are supported:

- **profile none**

This profile represents the existing system capabilities. This profile is indicated by the omission of the profile parameter in the BOF.

- **profile A**

This profile is intended for generic deployment scenarios, IP forwarding and MPLS switching use-cases.

- **profile B**

This profile is intended for applications requiring high packet manipulation and processing (for example, NAT and IPsec).

Use the following command to configure the profile:

- **MD-CLI**

```
bof system profile
```

- **classic CLI**

```
bof system-profile
```

When changing between system profiles, it is mandatory to remove all configuration commands for features that are not supported in the target system profile before rebooting the system, otherwise the reboot fails at the unsupported configuration command on startup.

On the 7705 SAR Gen 2, the following conditions apply about the profile parameter:

- The default system profile is **none** when the parameter is omitted.
- The parameter can be configured to either **profile-a** or **profile-b**.
- If the parameter is configured to an invalid value, it is ignored and profile **none** is used by the system.

Use the following command to display the BOF system profile:

- **MD-CLI**

```
admin show configuration bof | match profile
```

- **classic CLI**

```
show bof | match system-profile
```

The BOF system profile used by the system when it booted can be seen in the boot messages (using the **show boot-messages** command), which display the BOF read when rebooting.

Use the following command to display the system profile that is in use on the system.

```
show chassis | match "System Profile"
```

3.11 Configuring the Boot Options File with CLI

This section provides information about configuring BOF parameters with CLI.

3.11.1 Basic BOF configuration

The parameters which specify the location of the image filename that the router tries to boot from and the configuration file are in the BOF.

The most basic BOF configuration should include the following:

- primary address
- primary image location
- primary configuration location

The following is an example of a basic BOF configuration.

Example: MD-CLI

```
[ ]
A:admin@node-2# admin show configuration bof
# TiMOS-B-22.2.R1 both/x86_64 Nokia 7705 SAR Copyright (c) 2000-2022 Nokia.
# All rights reserved. All use subject to applicable license agreements.
# Built on Sat Feb 26 15:31:00 PST 2022 by builder in /builds/c/222B/R1/panos/main/sros
# Configuration format version 22.2 revision 0

# Generated 2022-03-07T17:08:41.4+00:00 by admin from Console

bof {
  configuration {
    primary-location "cf3:\config.cfg"
  }
  console {
    speed 115200
  }
  dns {
    domain "example.com"
    primary-server 10.251.72.68
    secondary-server 10.251.10.29
  }
  image {
    primary-location "cf3:\timos\"
  }
  li {
    local-save false
    separate false
  }
  license {
    primary-location "cf3:\license.txt"
  }
  port "management" {
    autonegotiate true
  }
  router "management" {
    interface "management" {
      cpm active {
        ipv4 {
          ip-address 192.168.189.52
          prefix-length 24
        }
      }
      cpm standby {
      }
    }
    static-routes {
      route 192.168.0.0/16 {
        next-hop 192.168.189.1
      }
      route 172.16.0.0/16 {
        next-hop 192.168.189.1
      }
    }
  }
  system {
    fips-140 false
    persistent-indices true
  }
}
```

```
# Finished 2022-03-07T17:09:40.4+00:00
```

Example: classic CLI

```
A:node-2# show bof
=====
BOF (Memory)
=====
primary-image      ftp://*:~@192.168.15.1/./images/
primary-config     ftp://*:~@192.168.15.1/./images/dut-a.cfg
address            192.168.189.53/16 active
address            192.168.189.54/16 standby
static-route       192.0.2.0/24 next-hop 192.0.2.254
static-route       192.168.0.0/16 next-hop 192.0.2.254
static-route       192.168.10.10/16 next-hop 192.0.2.254
autonegotiate
duplex              full
speed              100
wait               3
persist             off
console-speed      115200
=====
```

3.11.2 Common configuration tasks

This sections describes basic system tasks that must be performed to configure BOF.

For more information about hardware installation and initial router connections, see the specific hardware installation guide.

3.11.2.1 Searching for the BOF

The BOF should be on the same drive (cf3:) as the bootstrap image file. If the system cannot load or cannot find the BOF, the system checks whether the boot sequence was manually interrupted.

3.11.2.2 Accessing the CLI

To access the CLI to configure the software for the first time, perform the following steps:

- When the power to the chassis is turned on, the SR OS software automatically begins the boot sequence.
- When the boot loader and BOF image and configuration files are successfully located, establish a router connection (console session).

3.11.2.3 Console connection

To establish a console connection, you need the following:

- a ASCII terminal or a PC running terminal emulation software set to the parameters shown in the following table
- a standard serial cable connector for connecting to an RS232 port (provides an RJ-45 connector)

Table 15: Console configuration parameter values

Parameter	Value
Baud rate	115200
Data bits	8
Parity	None
Stop bits	1
Flow control	None

3.11.2.4 Configuring BOF encryption

The BOF contents are encrypted using AES256 and authenticated and hashed using SHA256.

Use the following command to configure BOF encryption:

- **MD-CLI**

```
bof configuration encrypt
```

- **classic CLI**

```
bof encrypt
```

3.11.2.5 Configuring the BOF interactive menu password

Access to the BOF interactive menu can be controlled using a password.

Use the following command to configure a BOF interactive menu password:

- **MD-CLI**

```
bof configuration password
```

- **classic CLI**

```
bof password
```

The password can be in one of the following formats:

- a plaintext string between 8 and 32 characters; the plaintext string cannot contain embedded nulls or end with "hash", "hash2", or "custom"



Caution: When entering the password in plaintext, ensure that the password is not visible to bystanders.

- a hashed string between 1 and 64 characters; the selected hashing scheme can be hash, hash2, or custom



Note: The hash2 encryption scheme is node-specific and the password cannot be transferred between nodes.

3.11.2.6 Configuring configuration file encryption

The configuration file contents can be encrypted using AES256 or SHA256.

Use the following command to configure a configuration file encryption key:

- **MD-CLI**

```
bof configuration encryption-key
```

- **classic CLI**

```
bof encryption-key
```

When configuring an encryption key, the key can be in one of the following formats:

- a plaintext string between 8 and 32 characters; the plaintext string cannot contain embedded nulls or end with "hash", "hash2", or "custom"



Caution: When entering the encryption key in plaintext, ensure that the key is not visible to bystanders.

- a hashed string between 1 and 64 characters; the selected hashing scheme can be hash, hash2, or custom



Note: The hash2 encryption scheme is node-specific and the key cannot be transferred between nodes.



Caution: In model-driven configuration mode with incremental saved configuration files enabled, the **admin save** command must be executed after changing configuration file encryption keys to ensure that a complete saved configuration file is saved with the new encryption key. After changing the encryption key, previously saved configuration files are no longer readable or loadable with the following command:

- **MD-CLI**

```
rollback
```

- **classic CLI**

```
admin rollback
```



Caution: Previously saved unencrypted configuration files, including incremental saved configuration files, are not automatically removed and must be removed manually.

3.11.3 Autoconfigure

When autoconfigure is enabled, the router performs a DHCP discovery or solicit (IPv6) to get the IP address of the out-of-band (OOB) management port.

The OOB management port can support a DHCP client for IPv4, IPv6, or dual stack. For dual stack, both IPv4 and IPv6 DHCP are configured. When the offer for either of the address families arrives, the management port is configured with the IP address in the offer. Eventually, both offers arrive and the management port is configured with both address families.

When a DHCP client is configured using autoconfigure, all image and license files should be placed and loaded from the CF. The configuration file could be loaded from the network, but Nokia recommends that the config file be on the CF as well. The configuration file is not loaded until the DHCP client offer is received and programmed successfully for the management port IP address, or the DHCP client timeout is expired.

3.11.3.1 Autoconfigure restrictions

When autoconfigure is enabled, a static IP address or static route cannot be configured in the BOF.

Similarly, a DNS server cannot be configured in the BOF, and only the DNS server provided by the DHCP offer can be used to resolve URLs.

The option 15 DNS domain name is not supported. The user can configure the DNS domain in the BOF so that the domain is not blocked when autoconfigure is used. Otherwise, the user must use the absolute URL with the hostname and domain included.

3.11.3.2 DHCP discovery of MAC addresses

When autoconfigure is used on redundant CPM chassis, the DHCP discovery uses the chassis MAC address. Only the active CPM performs a DHCP discovery and not the inactive CPM. When the offer arrives, the node uses that IP and the chassis MAC as addresses for management. Consequently, the inactive CPM is not reachable by the network, because it has no separate IP address. On activity switch, the inactive CPM inherits the active IP and chassis MAC.

For non-redundant CPMs, the management port MAC is used.



Note: The router must be rebooted when enabling autoconfigure for the first time to ensure that the CPM card uses the chassis MAC address.

3.11.3.3 IPv6 DUID

The SR OS supports type 2 DUID (link local), which is set to the chassis serial number. Type 3 (enterprise) is set to the chassis MAC address. Type 1 is not supported.

For type 2 DUID, the SR OS sends the Nokia Enterprise ID as the second byte of the DUID, followed by the chassis serial number. The first byte is the DUID type code. The chassis serial number starts with capital ASCII letters, which ensures that the serial number is unique as an application ID in the SR OS IPv6 DHCP application domain.

DUID type codes are as follows:

- DHCP6C_DUID_ENT_ID__IPSEC_IPV4ADDR - 1
- DHCP6C_DUID_ENT_ID__IPSEC_ASN1DN - 2
- DHCP6C_DUID_ENT_ID__IPSEC_FQDN - 3
- DHCP6C_DUID_ENT_ID__IPSEC_USER_FQDN - 4
- DHCP6C_DUID_ENT_ID__IPSEC_IPV6ADDR - 5
- DHCP6C_DUID_ENT_ID__IPSEC_ASN1GN - 6
- DHCP6C_DUID_ENT_ID__IPSEC_KEYID - 7
- DHCP6C_DUID_ENT_ID__WLAN_GW - 8
- DHCP6C_DUID_ENT_ID__AUTOBOOT - 9
- DHCP6C_DUID_ENT_ID__ZTP_BOF_AUTOP - Capital letters in ASCII

3.11.3.4 IPv6 DHCP RAs

An IPv6 DHCP offer does not have an IP prefix within the offer, unlike an IPv4 DHCP offer. The IPv6 prefix is usually obtained from the IPv6 Route Advertisement (RA) arriving from the upstream router. For ZTP, the SR OS is a host and assigns a /128 prefix to the IPv6 address obtained from the DHCP offer. In addition, the SR OS supports the installation of IPv6 default and static routes from upstream routers using the IPv6 RA. Multiple upstream routers can respond to a route solicitation with their own RA. The SR OS installs all the routes advertised by the RA. If the same route is advertised by multiple upstream routers (next hops), the SR OS installs the route with the highest preference. The SR OS does not support ECMP when the same route is advertised from multiple next hops by multiple RAs.

To ensure that all the RAs are obtained before the auto-provisioning process is started for IPv6, the SR OS follows the RFC 4861 recommendation that the host (in this case, the SR OS) send a minimum of three route solicitations. This is to ensure that if a route solicitation is lost, at least one of the three would reach the upstream routers. Each route solicitation is followed by a 4 s timeout. If the first route solicitation is sent at T0, the second is sent at T0+4 s and the third is sent at T0+8 s. The upstream routers must respond to the route solicitation within 0.5 s. This means that the SR OS has all of the RAs and the routes within 8.5 s of the first route solicitation. Therefore, the SR OS waits for a maximum of 9 s to receive all RAs.

If the DHCPv6 timeout is less than 9 s, the DHCPv6 timeout is honored even for the RA wait time. If the node has received a single RA and DHCP offer, the process is considered a success. However, it is possible that not all the RAs have arrived on the node because the node has waited less than 9 s.

3.11.4 Service management tasks

This section describes the service management tasks and the system administration commands.

3.11.4.1 System administration commands in the classic CLI

For more information about the supported classic CLI commands, see the *7705 SAR Gen 2 Classic CLI Command Reference Guide*.

Use the following administrative commands to perform management tasks.

```
admin display-config
admin reboot
```

```
admin save
```

3.11.4.1.1 Viewing the current configuration

Use the following command to display the current configuration. The **detail** option displays all default values. The **index** option applies to the classic CLI and displays only the persistent indexes.

```
admin display-config
```

Use the following command to display context-level information.

```
info detail
```

The following example shows a configuration file for the 7705 SAR Gen 2.

Example

```
# TiMOS-B-25.3.R1.I8049 both/x86_64 Nokia 7705 SAR Gen 2 Copyright (c) 2000-2025 Nokia.
# All rights reserved. All use subject to applicable license agreements.
# Built on Mon Mar 3 02:15:04 UTC 2025 by builder in /builds/00/I8049/panos/main/srux
# Configuration format version 25.3 revision 0

# Generated Fri Mar 14 22:33:00 2025 UTC

exit all
configure
#-----
echo "System Configuration"
#-----
  system
    name "7705-1"
    config-backup 5
    boot-good-exec "ftp://test.*.*./images/env.cfg"
    switchover-exec "ftp://test.*.*./images/env.cfg"
    management-interface
      cli
        md-cli
        no auto-config-save
      exit
    exit
  yang-modules
    no nokia-combined-modules
    nokia-submodules
  exit
exit
netconf
  no auto-config-save
exit
time
  ntp
    no shutdown
  exit
  sntp
    shutdown
  exit
  zone UTC
exit
exit
...
```

```
#-----
echo "System Configuration Mode Configuration"
#-----
    system
        management-interface
            configuration-mode mixed
        exit
    exit

exit all

# Finished Fri Mar 14 22:33:04 2025 UTC
```

3.11.4.1.2 Modifying and saving a configuration

If you modify a configuration file, the changes remain in effect only during the current power cycle unless a **save** command is executed. Changes are lost if the system is powered down or the router is rebooted without saving:

- Specify the file URL location to save the running configuration. If a destination is not specified, the files are saved to the location where the files were found for that boot sequence. The same configuration can be saved with different filenames to the same location or to different locations.
- The **detail** option adds the default parameters to the saved configuration.
- The **index** option forces a save of the index file.
- Changing the active and standby addresses without reboot standby CPM may cause a boot-env sync to fail.

Example: Saving the BOF configuration

```
A:node-2# bof save
Writing configuration to cf3:/bof.cfg ... OK
Completed.
```

Example: Saving the system configuration

```
A:node-2# admin save
Writing configuration to cf3:/config.cfg
Saving configuration ... OK
Completed.
```



Note: If the **persist** option is enabled and the **admin save** command is executed with an FTP path used as the file URL, two FTP sessions simultaneously open to the FTP server. The FTP server must be configured to allow multiple sessions from the same login; otherwise, the configuration and index files will not be saved correctly.

3.11.4.1.3 Deleting BOF parameters

You can delete specific BOF parameters. The changes remain in effect only during the current power cycle unless a **save** command is executed. Changes are lost if the system is powered down or the router is rebooted without saving.

Deleting a BOF address entry is not allowed from a remote session.

Use the **no** form of following commands to remove and save BOF configuration parameters.

```
bof address <ip-prefix/ip-prefix-length> [<cpm>]
bof autonegotiate
bof console-speed <baud-rate>
bof dns-domain <dns-name>
bof duplex <duplex>
bof ip-mtu <octets>
bof li-local-save
bof li-separate
bof license-file <file-url>
bof persist {on|off}
bof primary-config <file-url>
bof primary-dns <ip-address>
bof primary-image <file-url>
bof secondary-config <file-url>
bof secondary-dns <ip-address>
bof secondary-image <file-url>
bof speed <speed>
bof static-route <ip-prefix/ip-prefix-length> next-hop <ip-address>
bof system-base-mac <mac-address>
bof system-profile <profile>
bof tertiary-config <file-url>
bof tertiary-dns <ip-address>
bof tertiary-image <file-url>
bof wait <seconds>
```

Example: Saving BOF configuration parameters

```
A:node-2# bof save
Writing configuration to cf3:/bof.cfg ... OK
Completed.
```

3.11.4.1.4 Saving a configuration to a different filename

Save the current configuration with a unique filename to have additional backup copies and to edit parameters with a text editor. You can save your current configuration to an ASCII file.

The following example shows saving a configuration to a different location.

Example: Using the admin save command

```
A:node-2>admin save cf3:\testABC.cfg
Writing configuration to cf3:\testABC.cfg
Saving configuration ... OK
Completed.
```

3.11.4.1.5 Rebooting

When an **admin>reboot** command is issued, routers with redundant CPM are rebooted as well as the XMAs, XCMs, and IOMs. Changes are lost unless the configuration is saved. Use the **admin>save file-url** command to save the current configuration. If no command line options are specified, the user is prompted to confirm the reboot operation.

The following example shows a reboot.

Example

```
A:node-2>admin# reboot
Are you sure you want to reboot (y/n)? y
```

3.11.4.1.6 Setting the MTU value for the management port

You can configure the MTU for IP packets transmitted out the interface of the management router instance associated with the management port. The command applies to the SR OS, however, it does not necessarily apply during the boot loader processing.

The operational MTU for the port is set to the lesser of the values configured with the **ip-mtu** command and the management port MTU. For example, with the port MTU fixed at 1514 bytes and an Ethernet header size of 14 bytes, the MTU of the management port is 1500 bytes (the default operational IP MTU).

If the interface supports IPv6 packets, the command value must be set to 1280 or higher, in accordance with RFC 2460, *Internet Protocol, Version 6 (IPv6) Specification*. Use the following command to configure the MTU for IP packets transmitted out the interface of the management router instance.

```
bof ip-mtu
```

3.11.4.2 System administration commands in the MD-CLI

For more information about the supported MD-CLI commands, see the *7705 SAR Gen 2 MD-CLI Command Reference Guide*.

Use commands in the following context to perform management tasks.

```
admin
```

3.11.4.2.1 Viewing the current configuration

The **admin show configuration** command displays the current configuration for a specified configuration region (the default region is **configure**). The **booted** and **cflash-id** options are valid only for the **bof** configuration region.

Example: Detailed show output of BOF configuration file

The following example shows a BOF configuration file with the **detail** option to display all default and unconfigured values and the **units** option to show units where applicable.

```
A:admin@node-2# admin show configuration bof units detail
# TiMOS-B-22.10.R1 both/x86_64 Nokia 7705 SAR Copyright (c) 2000-2022 Nokia.
# All rights reserved. All use subject to applicable license agreements.
# Built on Sun Oct 30 14:49:55 PDT 2022 by builder in /builds/c/2210B/R1/panos/main/sros
# Configuration format version 22.10 revision 0
# Generated 2023-01-12T16:57:57.6-05:00 by admin from Console

bof {
  configuration {
    primary-location "cf3:\config.cfg"
    ## secondary-location
    ## tertiary-location
```

```
}
console {
    speed 115200 bps
    wait-time 3 seconds
}
dns {
    ## domain
    ## primary-server
    ## secondary-server
    ## tertiary-server
}
image {
    primary-location "cf3:\timos\"
    ## secondary-location
    ## tertiary-location
}
li {
    local-save false
    separate false
}
license {
    primary-location "cf3:\license.txt"
}
port "management" {
    autonegotiate true
    duplex full
    speed 100 megabps
}
router "management" {
    interface "management" {
        ## ip-mtu
        cpm active {
            ipv4 {
                ip-address 192.168.189.52
                prefix-length 24
            }
            ## ipv6
        }
        cpm standby {
            ## ipv4
            ## ipv6
        }
    }
    static-routes {
        route 192.168.0.0/16 {
            next-hop 192.168.189.1
        }
        route 172.16.0.0/16 {
            next-hop 192.168.189.1
        }
    }
}
system {
    ## base-mac-address
    fips-140 false
    ## gateway-role
    persistent-indices true
    ## profile
}
}
# Finished 2023-01-12T16:57:57.6-05:00
```

3.11.4.2.2 Modifying BOF parameters

BOF parameters can be modified via a BOF session in exclusive, private, or read-only configuration mode in the MD-CLI. The same configuration management commands that are available in the configure region are available in the bof region.



Note: Changing the active and standby addresses without rebooting the standby CPM may cause synchronization with the **boot-env** option to fail.
Deleting a BOF address entry is not allowed from a remote session.

Example

```
[/]
A:admin@node-2# bof exclusive
INFO: CLI #2060: Entering exclusive configuration mode
INFO: CLI #2061: Uncommitted changes are discarded on configuration mode exit

[ex:/bof]
A:admin@node-2# ?

configuration      + Enter the configuration context
console            + Enter the console context
dns                + Enter the dns context
image              + Enter the image context
li                 + Enter the li context
license            + Enter the license context
port               + Enter the port list instance
router             + Enter the router list instance
system             + Enter the system context
```

See the *7705 SAR Gen 2 MD-CLI Command Reference Guide* and the *7705 SAR Gen 2 MD-CLI User Guide* for more information.

3.11.4.2.3 Saving a configuration

Configuration changes are lost if the system is powered down or the router is rebooted before the changes are saved. If the URL location to save the running configuration is not specified, the files are saved to the location where the files were found for the boot sequence. The same configuration can be saved with different filenames to the same location or to different locations.

Changing the active and standby addresses without rebooting the standby CPM may cause synchronization with the **boot-env** option to fail.

The following command saves the running configuration for the configure region. If no URL is specified, the configuration is saved to the `config.cfg` file.

```
admin save
```

Example: Configuration save output

```
[admin]
A:admin@node-2# save
Writing configuration to cf3:\config.cfg
Saving configuration OK
Completed.
```

The BOF configuration is saved to `cf3:\bof.cfg` with every **commit** command.

Example

The BOF configuration can be manually saved to a backup file on a server or to a different location, as shown in the following example.

```
[ ]
A:admin@node-2# admin save bof ftp://10.9.236.68/backup/node-2/bof.cfg
Writing configuration to ftp://10.9.236.68/backup/node-2/bof.cfg OK
Completed.
```

Example

The following example saves the BOF configuration to a file, named `testbof.cfg` on `cf3:`.

```
[ ]
A:admin@node-2# admin save bof testbof.cfg
Writing configuration to cf3:\testbof.cfg OK
Completed.
```



Note: The BOF configuration file is saved in classic CLI format.

3.11.4.2.4 Rebooting

When a **reboot** command is issued, routers with redundant CPM are rebooted as well as the XMAAs, XCMs, and IOMs. If the **now** option is not specified, the user is prompted to confirm the reboot operation.

3.11.4.2.5 Setting the MTU value for the management port

The following command configures the MTU for IP packets transmitted out the interface of the management router instance associated with the management port.

```
bof router "management" interface "management" ip-mtu
```

The command applies to the SR OS but does not necessarily apply during the boot loader processing.

The operational MTU for the port is set to the lesser of the values configured with the **ip-mtu** command and the management port MTU. For example, with the port MTU fixed at 1514 bytes and an Ethernet header size of 14 bytes, the MTU of the management port is 1500 bytes (the default operational IP MTU).

If the interface supports IPv6 packets, the command value must be set to 1280 or higher, in accordance with RFC 2460 *Internet Protocol, Version 6 (IPv6) Specification*.

Example

```
[ex:bof]
A:admin@node-2# router "management" interface "management" ip-mtu ?

ip-mtu <number>
<number> - <512..9786> - bytes

Interface IP MTU
```

Note: The new value of this element takes effect when the candidate is committed.

4 Debug configuration

The **debug** configuration commands enable detailed debugging information for various protocols.

4.1 Debug configuration in the classic CLI

The **debug** commands in the classic CLI are available by entering the **debug** configuration context.

Debugging configuration is not persistent across CPM switchovers or router reboots. The **show debug** command displays debugging information.

Example: Show debug information

```
A:node-2# show debug
debug
  system
  netconf info
  exit
exit
```

The **admin debug-save** command saves the debugging configuration to `config.dbg` at the BOF **primary-config** location if a URL is not specified.

Example: Save debug configuration

```
A:node-2# admin debug-save
Writing configuration to cf3:\config.dbg
Saving configuration OK
Completed.
```

For a description of individual **debug** commands, see the *7705 SAR Gen 2 Classic CLI Command Reference Guide*.

4.1.1 Logging debug events in the classic CLI

The following is an example configuration for debug events that are stored in destination CLI log identifier 7. The log entries wrap at 50 entries (the configured value of **cli**).

Example: Configuration for stored debug events

```
A:node-2>config>log# log-id 7
A:node-2>config>log>log-id$ from debug-trace
A:node-2>config>log>log-id$ to cli 50
A:node-2>config>log>log-id$ info
-----
      from debug-trace
      to cli 50
      no shutdown
-----
```

After the log is configured, execute the following **tools** command in the CLI session that is intended to display output of the debug events. See the *7705 SAR Gen 2 Clear, Monitor, Show, Tools CLI Command Reference Guide* for more information about the **tools** command.

Example: Subscribe to debug log output to the CLI session

```
A:node-2# tools perform log subscribe-to log-id 7
```

Debug events can be displayed using the **show log** command and cleared using the **clear log** command.

Example: Unsubscribe from debug log output to the CLI session

```
A:node-2# show log log-id 7
=====
Event Log 7 log-name 7
=====
Description : (Not Specified)
Log contents [size=50 next event=2 (not wrapped)]

---snip---

A:node-2# clear log log-id 7
```

The following is an example of terminating the output of the logs to the CLI session using the **unsubscribe-from** command.

Example: Unsubscribe from debug log output to the CLI session

```
A:node-2# tools perform log unsubscribe-from log-id 7
```

4.2 Debug configuration in the MD-CLI

The **debug** commands in the MD-CLI are available in an exclusive, private, or read-only session using the explicit or implicit configuration mode. The same configuration management commands that are available in the configure region are available in the debug region.

Debugging configuration is not persistent across router reboots. Use the following command to display debugging information.

```
admin show configuration debug
```

The command displays debugging information and supports all configuration display formats, datastores, and output formats that are supported for other regions.

Example: Display debug information

```
[/]
A:admin@node-2# admin show configuration debug
# TiMOS-B-22.2.R1 both/x86_64 Nokia 7750 SR Copyright (c) 2000-2022 Nokia.
# All rights reserved. All use subject to applicable license agreements.
# Built on Sat Feb 26 15:31:00 PST 2022 by builder in /builds/c/222B/R1/panos/main/sros
# Configuration format version 22.2 revision 0

# Generated 2022-03-07T16:51:54.1+00:00 by admin from Console
debug {
```



```

system {
    management-interface {
        netconf info
    }
}

# Finished 2022-03-07T16:51:54.1+00:00

```

The **admin save debug** command saves the debugging configuration to `debug.cfg` at the following location if a URL is not specified.

```
bof configuration primary-location
```

Example: Save debug configuration

```

[/]
A:admin@node-2# admin save debug
Writing configuration to cf3:\debug.cfg
Saving configuration OK
Completed.

```

For descriptions of individual **debug** commands, see the *7705 SAR Gen 2 MD-CLI Command Reference Guide*.

4.2.1 Logging debug events in the MD-CLI

Example: Configuring a CLI log for debug events

The following is an example of a configuration for debug events that are stored in destination CLI log identifier 7. The log entries wrap at 50 entries (the configured value of **max-entries**).

```

*(ex)[configure log]
A:admin@node-2# log-id 7

*(ex)[configure log log-id "7"]
A:admin@node-2# source debug

*(ex)[configure log log-id "7"]
A:admin@node-2# destination cli max-entries 50

*(ex)[configure log log-id "7"]
A:admin@node-2# info
    source {
        debug true
    }
    destination {
        cli {
            max-entries 50
        }
    }

```

Example: Subscribing to a CLI log

After the **commit** command is issued to include the log in the running configuration, the following **tools** command can be executed in the CLI session that is intended to display output of the debug

events. See the *7705 SAR Gen 2 Clear, Monitor, Show, Tools CLI Command Reference Guide* for more information about the **tools** command.

```
[/]
A:admin@node-2# tools perform log subscribe-to log-id 7
```

Example: Displaying and clearing debug events

Debug events can be displayed using the **show log** command and cleared using the **clear log** command.

```
[/]
A:admin@node-2# show log log-id 7
=====
Event Log 7 log-name 7
=====
Description : (Not Specified)
Log contents [size=50 next event=2 (not wrapped)]
...

[/]
A:admin@node-2# clear log log-id 7
```

4.3 Debug configuration in mixed and model-driven mode

When debugging is configured in mixed or model-driven management mode, the following usage guidelines apply.

If the commands are available in the MD-CLI, the MD-CLI commands must be used to configure debugging:

- The classic CLI cannot be used.
- Debug configuration commands entered in the MD-CLI are only displayed in the MD-CLI **info** and in the following command output.

```
admin show configuration debug
```

- Debug configuration entered in the MD-CLI can be saved to `debug.cfg` or a file URL with the **admin save debug** command.
- Debug configuration commands entered in the MD-CLI are not displayed in the classic CLI **show debug** output.



Note: References must be configured with the MD-CLI names, not the classic CLI IDs. For example, the MD-CLI VPRN service name, not the classic CLI VPRN service ID.

If the commands are not available in the MD-CLI, the classic CLI must be used to configure debugging:

- The MD-CLI cannot be used.
- Debug configuration commands entered in the classic CLI are only displayed in the classic CLI **show debug** output.

- Debug configuration entered in the classic CLI can be saved to `config.dbg` or a file URL with the **admin debug-save** command.
- Debug configuration commands entered in the classic CLI are not displayed in the MD-CLI **info** or in the following command output.

```
admin show configuration debug
```

The user must manually remove the classic and model-driven debug configuration before changing the management interface configuration mode from model-driven to mixed mode. The system automatically removes the classic and model-driven debug configuration during all other mode switches.

5 Secure boot

The SR OS Secure Boot ensures that the software executed by the system is trusted and originated from Nokia IP Routing.

At every boot of the control card, each step in the boot process verifies the digital signature of the next software element to boot for integrity and authenticity up to the SR OS operating system images. This boot sequence forms the chain of trust for Secure Boot.

Software image signatures use RSA-4096 keys and SHA-384 hashes.

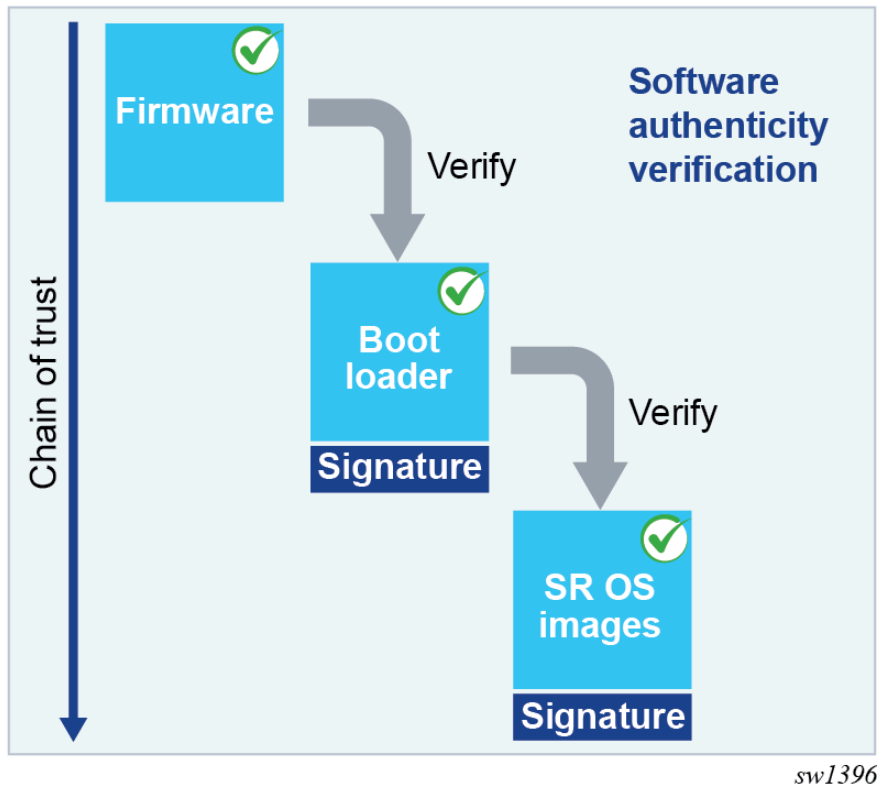
The Secure Boot chain is rooted in the platform CPM firmware based on UEFI specifications. As such, Nokia Platform Key, Key Exchange Key, allowed and disallowed databases are provisioned when Secure Boot is activated to perform the required signature verification.

Firmware updates are also digitally signed and verified using the same principle. The signature verification of a firmware update is performed at boot time by the existing firmware before the firmware update can proceed.

5.1 Secure Boot chain

Secure Boot is enabled by default and the Secure Boot chain of trust for SR OS platforms can be visualized with the following diagram.

Figure 5: Secure boot chain of trust



5.2 Operational commands and logs

This section describes the following:

- Secure boot state
- Software update process
- Update Secure Boot variables

5.2.1 Secure Boot state

Secure Boot and UEFI variables Secure Boot keys status is available per CPM.

Use the following command to display Secure Boot state information.

```
show card A detail
```

Output example

```
Hardware Data
Secure boot status      : enabled
```

```
UEFI variables status      : ok
```

where

- Secure Boot status — indicates if Secure Boot is enabled or disabled
- UEFI variables status — indicates if Secure Boot variables need updating

The system records at every boot in the security log if Secure Boot is enabled or disabled per CPM. The following is an example of such a log message.

```
24 2023/05/17 06:09:03.140 EDT MAJOR: SECURITY #2241 Base Card A
"CPM A has booted with a secure-boot status of enabled"
```

Secure Boot UEFI variables can be obtained per CPM card using the following command:

- **MD-CLI**

```
perform system security secure-boot show uefi-variables card
```

- **classic CLI**

```
tools dump system security secure-boot uefi-var card
```

The command displays the following x509 certificates and SHA-256 hash UEFI variables:

- Platform Key (PK)
- Key Exchange Key (KEK)
- Allowed Database (DB)
- Disallowed Database (DBx)

5.2.2 Software update

After Secure Boot is enabled, and before upgrading to a new software release, the user must validate that the new software image is properly signed. The main reason for this additional verification on systems with Secure Boot enabled is because the system only boots Nokia-signed software images and does not boot unsigned or improperly signed images.

Use the following command to validate the signature of the TiMOS *.tim images contained in the **software-image** *url* location referenced in the command. This verification includes `cpm.tim`, `iom.tim`, `support.tim`, `both.tim`, `kernel.tim`, as well as the `boot.ldr` if present in CF3 directory.

```
admin system security secure-boot validate software-image url
```

5.2.3 Update Secure Boot variables

The system supports Secure Boot UEFI key updates and revocation using the following commands.

```
admin system security secure-boot update-key
admin system security secure-boot revoke-key
```

6 System management

This chapter provides information about configuring basic system management parameters.

6.1 System management commands

System management commands allow you to configure basic system management functions, such as the system name, contact, router location and coordinates, naming objects, and CLI code, as well as time zones, Network Time Protocol (NTP), Simple Network Time Protocol (SNTP) synchronization, Precision Time Protocol (PTP), and CRON.

On SR OS routers, it is possible to query the DNS server for IPv6 addresses. By default, the DNS names are queried for A-records only (address-preference is IPv4-only). If the address-preference is set to IPv6 first, the DNS server is queried for AAAA-records first, and if there is no successful reply, then A-records.

6.1.1 System information

This section describes the system information components.

6.1.1.1 Name

You can configure a name for the system device. The name is used in the prompt string. Only one system name can be configured. If multiple system names are configured the last one encountered overwrites the previous entry. Use the following command to configure the system name.

```
configure system name
```

6.1.1.2 Contact

Use the **contact** command to specify the name of a system administrator, IT staff member, or other administrative entity.

Use the following command to configure the contact.

```
configure system contact
```

6.1.1.3 Location

Use the **location** command to specify the location of the device. For example, enter the city, building address, floor, room number, and so on, where the router is located.

Use the following command to configure the location.

```
configure system location
```

6.1.1.4 Coordinates

You can optionally configure the GPS location of the device. If the string contains special characters (#, \$, spaces, and so on), the entire string must be enclosed within double quotes. Use the following command to configure the system coordinates.

```
configure system coordinates
```

6.1.1.5 Naming objects

Avoid configuring named objects with a name that starts with "_tmnx_" and with "_" in general.

6.1.1.6 Common language location identifier

A Common Language Location Identifier (CLLI) for the device is an 11-character standardized code string that uniquely identifies the geographic location of places and specific functional categories of equipment unique to the telecommunications industry. The CLLI code is stored in the Nokia Chassis MIB tmnxChassisCLLIcode object.

The CLLI code can be any ASCII printable text string of up to 11 characters.

6.1.1.7 DNS security extensions

DNS Security (DNSSEC) Extensions are now implemented in the SR OS, allowing operators to configure DNS behavior of the router to evaluate whether the Authenticated Data bit was set in the response received from the recursive name server and to trust the response, or ignore it.

6.1.2 System time

Routers are equipped with a real-time system clock for timekeeping purposes. When set, the system clock always operates on Coordinated Universal Time (UTC), but the software has options for local time translation as well as system clock synchronization.

6.1.2.1 Time zones

Setting a time zone in SR OS allows for times to be displayed in the local time rather instead of UTC. SR OS has both user-defined and system-defined time zones.

A user-defined time zone has a user-assigned name of up to four printable ASCII characters in length and is unique from the system-defined time zones. For user-defined time zones, the offset from UTC is configured as well as any summer time adjustment for the time zone.

SR OS includes multiple commands to control the presentation of times in either UTC or local time zone format. For a CLI session, the environment variable `time-display` may be set to indicate UTC or local time zone. This setting only affects time strings shown during that specific CLI session. A global setting of the following command can be used to control time strings for objects with larger scope than a single CLI session.

```
configure system time prefer-local-time
```

Time strings include the following:

- log filenames and log header information
- times in rollback information
- times in rollback and configuration files header information
- times related to CRON scripts
- times related to CRON scripts
- times in the event handler system
- times in NETCONF and gRPC date-and-time leafs

A separate control per log file controls the format of the time strings on the event recorded into the logs (separate from the log filename and header information). Use the following command to set these time strings.

```
configure log log-id time-format
```

The SR OS system-defined time zones are listed in the following table, which includes both time zones with and without daylight saving (summer) time adjustment.

Table 16: System-defined time zones and UTC offsets

Acronym	Time zone name	UTC offset
Europe:		
GMT	Greenwich Mean Time	UTC
BST	British Summer Time	UTC +1
IST	Irish Summer Time	UTC +1*
WET	Western Europe Time	UTC
WEST	Western Europe Summer Time	UTC +1
CET	Central Europe Time	UTC +1
CEST	Central Europe Summer Time	UTC +2
EET	Eastern Europe Time	UTC +2
EEST	Eastern Europe Summer Time	UTC +3
MSK	Moscow Time	UTC +3

Acronym	Time zone name	UTC offset
MSD	Moscow Summer Time	UTC +4
US and Canada:		
AST	Atlantic Standard Time	UTC -4
ADT	Atlantic Daylight Time	UTC -3
EST	Eastern Standard Time	UTC -5
EDT	Eastern Daylight Saving Time	UTC -4
ET	Eastern Time	Either as EST or EDT, depending on place and time of year
CST	Central Standard Time	UTC -6
CDT	Central Daylight Saving Time	UTC -5
CT	Central Time	Either as CST or CDT, depending on place and time of year
MST	Mountain Standard Time	UTC -7
MDT	Mountain Daylight Saving Time	UTC -6
MT	Mountain Time	Either as MST or MDT, depending on place and time of year
PST	Pacific Standard Time	UTC -8
PDT	Pacific Daylight Saving Time	UTC -7
PT	Pacific Time	Either as PST or PDT, depending on place and time of year
HST	Hawaiian Standard Time	UTC -10
AKST	Alaska Standard Time	UTC -9
AKDT	Alaska Standard Daylight Saving Time	UTC -8
Australia and New Zealand:		
AWST	Western Standard Time (for example, Perth)	UTC +8 hours
ACST	Central Standard Time (for example, Darwin)	UTC +9.5 hours
AEST	Eastern Standard/Summer Time (for example, Canberra)	UTC +10 hours
NZT	New Zealand Standard Time	UTC +12 hours
NZDT	New Zealand Daylight Saving Time	UTC +13 hours

6.1.2.2 NTP

NTP is the Network Time Protocol defined in RFC 1305, *Network Time Protocol (Version 3) Specification, Implementation and Analysis* and RFC 5905, *Network Time Protocol Version 4: Protocol and Algorithms Specification*. It allows for the participating network nodes to keep time more accurately and more importantly they can maintain time in a more synchronized fashion between all participating network nodes.

SR OS uses an NTP process based on a reference build provided by the Network Time Foundation. Nokia strongly recommends that the users review RFC 8633, *Network Time Protocol Best Current Practices*, when they plan to use NTP with the router. The RFC section "Using Enough Time Sources" indicates that using only two time sources (NTP servers) can introduce instability if they provide conflicting information. To maintain accurate time, Nokia recommends configuring three or more NTP servers.

NTP uses stratum levels to define the number of hops from a reference clock. The reference clock is considered to be a stratum-0 device that is assumed to be accurate with little or no delay. Stratum-0 servers cannot be used in a network. However, they can be directly connected to devices that operate as stratum-1 servers. A stratum-1 server is an NTP server with a directly-connected device that provides Coordinated Universal Time (UTC), such as a GPS or atomic clock.

The higher stratum levels are separated from the stratum-1 server over a network path, therefore, a stratum-2 server receives its time over a network link from a stratum-1 server. A stratum-3 server receives its time over a network link from a stratum-2 server.

SR OS routers normally operate as a stratum-2 or higher device. The router relies on an external stratum-1 server to source accurate time into the network. However, SR OS also allows for the use of the local PTP recovered time to be sourced into NTP. In this latter case, the local PTP source appears as a stratum-0 server and SR OS advertises itself as a stratum-1 server. Activation of the PTP source into NTP may impact the network NTP topology because the SR OS router is promoted to stratum-1.

SR OS router runs a single NTP clock which then operates NTP message exchanges with external NTP clocks. Exchanges can be made with external NTP clients, servers, and peers. These exchanges can be through the base, management, or VPRN routing instances.

NTP operates associations between clocks as either client or server, symmetric active and symmetric passive, or broadcast modes. These modes of operation are applied according to which elements are configured on the router. To run server mode, the operator must enable NTP server mode for the base and each needed VPRN routing instance. To run client mode, the operator must configure external servers. If both the local router and remote router are configured with each other as peers, then the router operates in symmetric active mode. If only one side of the association has peering configured, then the modes are symmetric passive. To operate using broadcast mode, interfaces must be configured to transmit as broadcast servers or receive as broadcast clients.

NTP server operation for both unicast and broadcast communication within a VPRN is configured within the VPRN (see "NTP Within a VPRN Service" in the *7705 SAR Gen 2 Layer 3 Services Guide: IES and VPRN*).



Note: NTP provides lightweight synchronization across a network for alignment of system time for logging purposes. NTP does not provide the high accuracy time needed for the on-air applications of the mobile base stations.

The following NTP elements are supported:

- **server mode**

In this mode, the node advertises the ability to act as a clock source for other network elements. The node, by default, transmits NTP packets in NTP version 4 mode.

- **authentication keys**

Authentication keys implement increased security support in carrier and other networks. Both DES and MD5 authentication are supported, as well as multiple keys.

- **operation in symmetric active mode**

This capability requires that NTP be synchronized with a specific node that is considered more trustworthy or accurate than other nodes carrying NTP in the system. This mode requires that a specific peer is set.

- **server and peer addressing using IPv6**

Both external servers and external peers may be defined using IPv6 or IPv4 addresses. Other features (such as multicast, broadcast) use IPv4 addressing only.

- **broadcast or multicast modes**

When operating in these modes, the node receives or sends using either a multicast (default 224.0.1.1) or a broadcast address. Multicast is supported only on the CPM MGMT port.

- **alert when NTP server is not available**

When none of the configured servers are reachable on the node, the system reverts to manual timekeeping and issues a critical alarm. When a server becomes available, a trap is issued indicating that standard operation has resumed.

- **NTP and SNTP**

If both NTP and SNTP are enabled on the node, then SNTP transitions to an operationally down state. If NTP is removed from the configuration or shut down, then SNTP resumes an operationally up state.

- **gradual clock adjustment**

As several applications (such as Service Assurance Agent (SAA)) can use the clock, and if determined that a major (128 ms or more) adjustment needs to be performed, the adjustment is performed by programmatically stepping the clock. If a minor (less than 128 ms) adjustment must be performed, then the adjustment is performed by either speeding up or slowing down the clock.

- To avoid the generation of too many events/trap the NTP module rates limit the generation of events/traps to three per second. At that point a single trap is generated that indicates that event/trap squashing is taking place.

6.1.2.3 CRON

The CRON feature supports periodic and date and time-based scheduling in SR OS. CRON can be used, for example, to schedule Service Assurance Agent (SAA) functions. CRON functionality includes the ability to specify scripts that need to be run, when they are scheduled, including one-time only functionality (one-shot), interval and calendar functions. Scheduled reboots, peer turn ups, service assurance agent tests and more can all be scheduled with CRON, as well as OAM events, such as connectivity checks, or troubleshooting runs.

CRON supports the schedule element. The schedule function configures the type of schedule to run, including one-time only (one-shot), periodic, or calendar-based runs. All runs are determined by month, day of month or weekday, hour, minute, and interval (seconds).

6.2 IP hashing as an LSR

It is now possible to include IP header in the hash routine at an LSR for the purpose of spraying labeled-IPv4 and labeled-IPv6 packets over multiple equal cost paths in ECMP in an LDP LSP and over multiple links of a LAG group in all types of LSPs.

A couple of configurable options are supported. The first option is referred to as the Label-IP Hash option and is designated in the CLI as **lbl-ip**. When enabled, the hash algorithm parses down the label stack and after it hits the bottom of the stack, it checks the next nibble. If the nibble value is four or six then it assumes it is an IPv4 or IPv6 packet. The result of the hash of the label stack, along with the incoming port and system IP address, is fed into another hash along with source and destination address fields in the IP packet's header. The second option is referred to as IP-only hash and is enabled in CLI using **ip-only**. It operates the same way as the Label-IP Hash method except the hash is performed exclusively on the source and destination address fields in the IP packet header. This method supports both IPv4 and IPv6 payload.

By default, MPLS packet hashing at an LSR is based on the whole label stack, along with the incoming port and system IP address. This method is referred to as the Label-Only Hash option and is enabled by entering **lbl-only**.

Use the following context to configure **lbl-only**, **lbl-ip**, and **ip-only** on a system-wide basis or override them on a per-IP-interface basis.

```
configure system load-balancing lsr-load-balancing
```

6.3 Auto-provisioning

Auto-provisioning is used to provision a node using an external DHCP server and file server. It is used to obtain a configuration file and an image file from an external server using an in-band mechanism. Auto-provisioning is not compatible with an out-of-band management port.

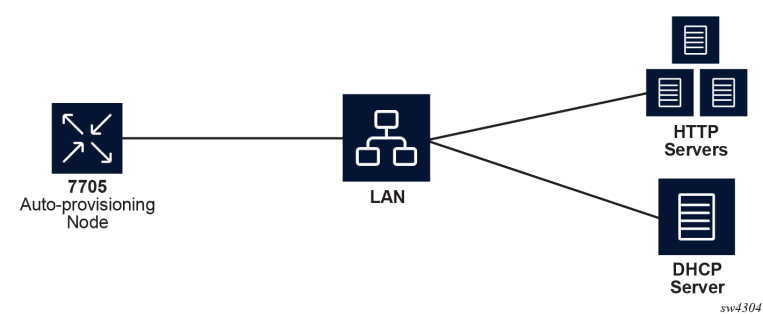
Before using auto-provisioning, the SR OS must be booted up and running the application image. In addition, it needs to have some minimum configuration before the auto-provision script is executed by the operator.

After the auto-provision application is triggered using a tools command, SR OS checks all operationally up ports without IP addresses and send DHCP discovery to these interfaces. The DHCP server needs to be configured with Option 67 and the user must provide the SR OS with the URL of a file server and the corresponding directory for the image.

[Figure 6: Example of a network with no DHCP relay](#) to [Figure 8: Example of a network with multiple subnets](#) describe scenarios in which auto-provisioning are used.

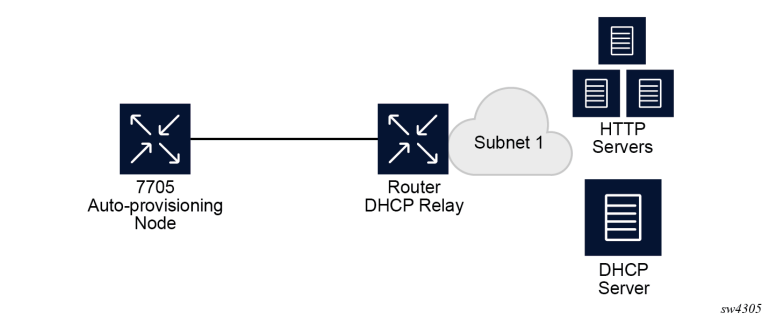
In [Figure 6: Example of a network with no DHCP relay](#), there is no DHCP relay and all IP addresses are assigned from a single pool.

Figure 6: Example of a network with no DHCP relay



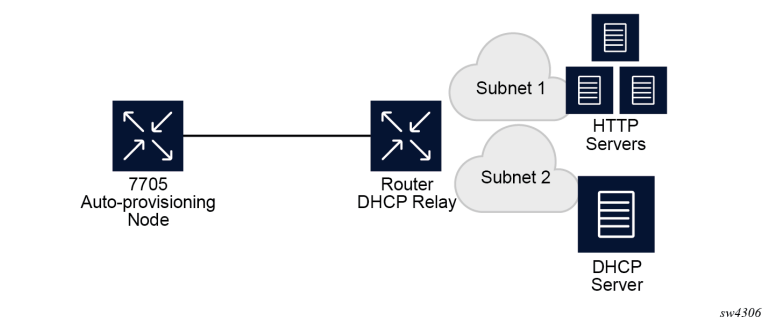
In [Figure 7: Example of a network with a DHCP relay](#), there is a DHCP relay which injects the Option 82 as a gateway address. The DHCP server is assigned the IP address from the pool dictated by the gateway address option 82. The DHCP server and HTTP server are in the same subnet. The DHCP offer has option 3 "router" which is used for a default gateway creation on the 7705 SAR Gen 2.

Figure 7: Example of a network with a DHCP relay



In [Figure 8: Example of a network with multiple subnets](#), all components are in different subnets. The DHCP relay adds Option 82 to the DHCP request as the gateway address which is used for pool selection. The DHCP server must add option 3 configured with the gateway address of the HTTP server.

Figure 8: Example of a network with multiple subnets



6.3.1 Auto-provisioning limits

The following are some configuration limits for auto-provisioning:

- A maximum of 12 Layer 3 interfaces are supported for auto-provisioning.
- Only IPv4 auto-provisioning is supported.
- It is highly recommended to only have a basic card, MDA, port, and interface configuration as described in this document and no additional static routes or IGP or BGP protocols when performing auto-provisioning because auto-provisioning installs default static routes that may be affected by any extra routing configuration.
- A maximum of 255 characters is supported for the remote URL (200 character maximum for the filepath, the rest for the main URL consisting of the protocol, login credentials, and host IP). A maximum of 200 characters is supported for the local URL. The local file or folder name must not exceed 99 characters.
- The maximum number of file pairs for each image/config record is 10.

6.3.2 Auto-provisioning process

In this process, the node detects operational ports, attempts to discover its IP address, and downloads the relevant files for provisioning.

1. The node sends a DHCP discovery request to the DHCP server using the out-of-band management port. If DHCP discovery is unsuccessful, the node reattempts it using the in-band management ports.
2. After DHCP discovery is successful, the DHCP server returns an IPv4 or IPv6 FTP or HTTP URL of a file server from which the node can retrieve provisioning information.
3. The node downloads the provisioning information and performs the auto-provisioning according to the specifications in the files.
4. After the node is successfully provisioned, it automatically reboots and becomes operationally up.

See [Provisioning files](#) for more information about the auto-provisioning process.

The SR OS can also initiate the auto-provisioning process using a **tools** command.

6.3.3 Auto-provisioning DHCP rules

The following are the DHCP rules in the auto-provisioning stage:

1. First, auto-provisioning walks through the interfaces with a configured port, where the port is in operational status up, one by one.
2. It sends a DHCP request to the first configured interface with a port up and no IP address configured.
 - If, on this interface, multiple DHCP offers arrives, only the first offer is sent to the auto-provisioning task and the other offers are ignored. This could occur if the node is on a LAN and multiple DHCP servers are connected to the interface.
 - The DHCP client has an exponential retry mechanism. If the DHCP offer does not arrive from the server, the client resends a DHCP request at 2, 4, 8, 32 and 64 s, with 64 s being the maximum timeout, If the 64 s timeout interval is reached, the DHCP client keeps retrying every 64 s. The

user can configure a timeout value. If no DHCP offer has arrived by this timeout value, the auto-provisioning process moves to the next interface.

- If the DHCP offer arrives on the port and the DHCP client task does not acknowledge the DHCP offer, for any reason, it disables the DHCP client and remove the IP from the port.
 - If the DHCP offer arrives on the port and the DHCP client acknowledges the offer, it sends the information to auto-provisioning. If auto-provisioning does not like the offer, because there is no Option 67, Option 67 is malformed, or for any other reason listed in [Auto-provisioning failure](#), the auto-provisioning process deconfigures the DHCP client and the DHCP client sends a DHCP release, and unassigns the IP address.
 - In case of failure, more information is displayed by the auto-provisioning process and the process moves to the next port that is up and does not have an IP address.
3. If auto-provisioning is successful using the offer and its option, the provisioning file download starts though the protocol dictated by Option 67.

The **auto-provisioning** command is CLI blocking. All information about the auto-provisioning process is displayed on the CLI and logged.

6.3.4 Auto-provisioning failure

Auto-provisioning fails for the following reasons:

- There is no Option 67.
- The Option 67 format is not acceptable to auto-provisioning.
- The format is a URL or DNS is not supported. There is a failure in the download provisioning file or the server is not reachable.
- There is failure in the download of the image or config file using the provisioning file information, for example, the server is not available, the wrong directory is listed, or the wrong credentials are given.
- The image or config fails to copy to the compact flash.
- The image or config fails to sync to the inactive CPM.
- The BOF does not point to the compact flash, for example, it is pointing to the network.

If the auto-provisioning procedure on this interface fails, then auto-provisioning does the following:

1. Displays information about the blocked CLI and in the log, describing the failure in detail.
2. Updates the DHCP task so the DHCP task can take the appropriate actions to release the IP address on the interface. This is done by sending a DHCP release for the DHCP ack received from the server.
3. Goes to the next interface with port up and no IP address.



Note: If no other interface with port up is found, the auto-provisioning task stops and a failure error is displayed on the CLI and in the log.

6.4 Administrative tasks

This section contains information to perform administrative tasks.

6.4.1 Saving configurations

Whenever configuration changes are made, the modified configuration must be saved so they are not lost when the system is rebooted.

Configuration files are saved by executing explicit command syntax which includes the file URL location to save the configuration file as well as options to save both default and non-default configuration parameters. Boot option file (BOF) parameters specify where the system should search for configuration and image files as well as other operational parameters during system initialization.

For more information about boot option files, see the Boot Options section.

6.4.2 Specifying post-boot configuration files

Two post-boot configuration extension files are supported and are triggered when either a successful or failed boot configuration file is processed. The **boot-bad-exec** and **boot-good-exec** commands specify URLs for the CLI scripts to be run following the completion of the bootup configuration. A URL must be specified or no action is taken.

For example, after a configuration file is successfully loaded, the specified URL can contain a nearly identical configuration file with specific commands enabled or disabled, or particular parameters specified and according to the script which loads that file.

6.4.3 Network timing

In Time Domain Multiplexed (TDM)-based networks (for example, SONET or SDH circuit-switched networks), the concept of network timing is used to prevent over-run or under-run issues where circuits are groomed (rebundled) and switched. Hardware exists in each node that takes a common clock derived from an internal oscillator, a specific receive interface, or special BITS interface and provides it to each synchronous interface in the system. Usually, each synchronous interface is allowed to choose between using the chassis-provided clock or the clocking recovered from the received signal on the interface. The clocking is used to drive the transmit side of the interface. The appropriate configuration at each node which defines how interface clocking is handled must be considered when designing a network that has a centralized timing source so each interface is operating in a synchronous manner.

The effect of timing on a network is dependent on the nature of the type of traffic carried on the network. With bit-wise synchronous traffic (traditional circuit-based voice or video), non-synchronous transmissions cause a loss of information in the streams affecting performance. With packet-based traffic, the applications expect and handle jitter and latency inherent to packet-based networks. When a packet-based network is used to carry voice or video traffic, the applications use data compression and elasticity buffering to compensate for jitter and latency. The network itself relies on appropriate Quality of Service (QoS) definitions and network provisioning to further minimize the jitter and latency the application may experience.

6.4.4 Power supplies

SR OS supports a **power-supply** command to configure the type and number of power supplies present in the chassis. The operational status of a power source is always displayed by the LEDs on the Control Processor/Switch Fabric Module (CP/SFM) front panel, but the power supply information must be explicitly

configured in order for a power supply alarm to be generated if a power source becomes operationally disabled.

6.5 System router instances

SR OS supports multiple Layer 3 router instances. These instances have their own IP addressing spaces and configuration options. Router instances are isolated from each other.

The following are the different types of router instances in SR OS:

- **Base**

All SR OS routers have the base router instance: the system created default router instance used to forward user IP traffic among router line card ports. Router interfaces (that is, network interfaces configured under **configure router [Base]**) and IES services and interfaces exist in the base router instance. The base router instance is identified in SNMP as vRtrType = baseRouter (1) and has a vRtrID of 1.

- **VPRN instances**

Another type of router instance is the set of operator configured VPRN services. Each VPRN service has a unique router instance. For more information about VPRN services and their associated router instances, see the *7705 SAR Gen 2 Layer 3 Services Guide: IES and VPRN*. VPRN router instances are identified in SNMP as vRtrType = vprn (2), and the vRtrID is dynamically allocated.

- **Special system router instances**

SR OS routers also support the following special router instances:

- **management**

The management router instance is a system created router instance that is used for management of the router. The management router instance is bound to CPM/CCM ports A/1 and B/1. This is a CPM router instance which cannot be renamed or deleted by an operator. The management router instance is identified in SNMP as vRtrType = vr(3), and the vRtrID is 4095.

- **vppls-management**

The vppls-management router instance is used for management of VPLS services. It is identified in SNMP as vRtrType = vr(3), and the vRtrID is 4094.

- **User created CPM router instances**

User created CPM router instances are user defined router instances that are mainly used with Ethernet ports on the CPM/CCM cards: CPM router instances only use CPM/CCM Ethernet ports as interfaces. CPM router instances have a user-defined name and are the only types of non-VPRN router instances that can be created by the user. User created CPM router instances are identified in SNMP as vRtrType = vr(3), and the vRtrID is dynamically allocated.

Some management protocols can use either the base routing instance (in-band) or the management routing instance (out-of-band). A listing of these protocols can be found in the CPM Filter: Protocols and Ports section of the *7705 SAR Gen 2 System Management Guide*. Unless otherwise stated in the detailed description of the protocol, when the server or client for the protocol is reachable via the management routing instance, those protocol messages use the management interface for the protocol communication.

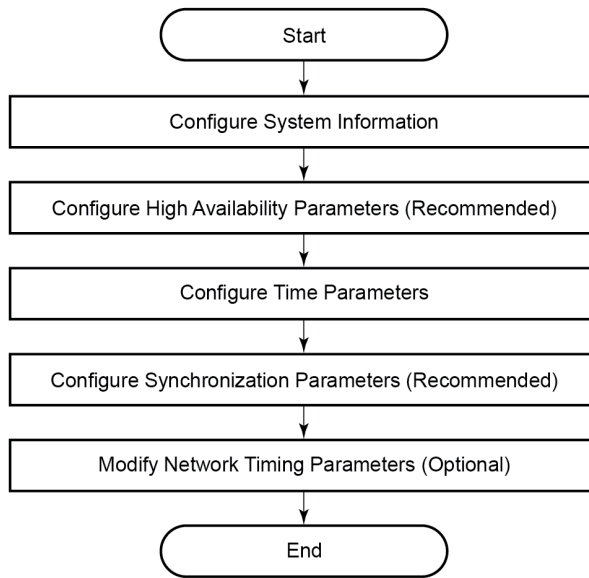
If BOF is set up with autoconfiguration and the DHCP server provides a general default route such as 0.0.0.0/0, with some protocols (like PCEP, TACACS+, RADIUS, and LDAP), Authentication, Authorization, Accounting (AAA) always prefers OOB over in-band connectivity. This is because these protocols prefer to

use the OOB management port first. If a matching route is not found, in-band is attempted. The static route provided by DHCP must be properly set to ensure the correct route preference is made by these protocols.

6.6 System configuration process overview

The following figure shows the process for basic system provisioning.

Figure 9: System configuration and implementation flow



7750_SR_Basics_27

6.7 Configuration notes

The system must be correctly initialized and the boot loader and BOF successfully executed to access the CLI.

6.8 Configuring system management features

This section provides information about configuring system management features.

6.8.1 Saving configurations

Whenever configuration changes are made, the modified configuration must be saved so the changes are not lost when the system is rebooted. The system uses the configuration and image files, as well as other operational parameters necessary for system initialization, according to the locations specified in the boot option file (BOF) parameters. For more information about BOFs, see the [System initialization and boot options](#) chapter of this manual:

Configuration files are saved by executing the **explicit** or **implicit** commands.

- An **explicit** save writes the configuration to the location specified in the **save** command (the file URL).
- An **implicit** save writes the configuration to the file specified in the primary configuration location.

If the **file-url** is not specified in the **save** command configuration, the system attempts to save the current configuration to the current BOF primary configuration source. If the primary configuration source (path and/or filename) changed since the last boot, the new configuration source is used.

The save command includes an option to save both default and non-default configuration (the **detail** option).

The **index** option specifies that the system preserves system indexes when a save command is executed, regardless of the persistent status in the BOF file. During a subsequent boot, the index file is read along with the configuration file. As a result, a number of system indexes are preserved between reboots, including the interface index, LSP IDs, path IDs, and so on. This reduces resynchronizations of the Network Management System (NMS) with the affected network element.

If the save attempt fails at the destination, an error occurs and is logged. The system does not try to save the file to the secondary or tertiary configuration sources unless the path and filename are explicitly named with the **save** command.

6.9 Basic system configuration

This section provides information to configure system parameters and provides configuration examples of common configuration tasks. The minimum system parameters that should be configured are:

- System time elements

Use the following command to display basic system information such as the system name, platform type, and so on.

```
show system information
```

6.10 Common configuration tasks

This section provides an overview of the CLI commands used to configure system parameters:

- [System information](#)
- [System time elements](#)

6.10.1 System information

This section describes the basic system information commands that configure the system name of the router, contact information, location (such as an address, floor, room number, and so on), CLI code, and global positioning system (GPS) coordinates.

6.10.1.1 System name

The device's system name is used in the prompt string. Only one system name can be configured; if multiple system names are configured, the last one overwrites the previous entry.

Use the following command to configure the system name.

```
configure system name
```

6.10.1.2 Contact

Use the **contact** command to specify the name of a system administrator, IT staff member, or other administrative entity.

Use the following command to configure the contact.

```
configure system contact
```

6.10.1.3 Location

Use the **location** command to specify the location of the device. For example, enter the city, building address, floor, room number, and so on, where the router is located.

Use the following command to configure the location.

```
configure system location
```

6.10.1.4 CLLI code

The Common Language Location Code (CLLI code) is an 11-character standardized geographic identifier that is used to uniquely identify the geographic location of a router.

Use the following command to configure the CLLI code.

```
configure system clii-code
```

6.10.1.5 GPS coordinates

Use the optional **coordinates** command to specify the GPS location of the device. If the string contains special characters (#, \$, spaces, and so on), the entire string must be enclosed within double quotes.

Use the following command to configure the coordinates.

```
configure system coordinates
```

6.10.2 System time elements

The system clock maintains time according to Coordinated Universal Time (UTC). Configure the time zone and summer time (daylight saving time) command options to correctly display time according to the local time zone.

6.10.2.1 Zone

The **zone** command sets the time zone and time zone UTC offset for the router. SR OS supports system-defined and user-defined time zones. The system-defined time zones and offsets are listed in [Table 16: System-defined time zones and UTC offsets](#).

Use the following command to set the time zone and time zone UTC offset.

```
configure system time zone
```

6.10.2.2 Summer (daylight saving) time

Configure the start and end dates and offset for summer (daylight saving) time to override system defaults or for user-defined time zones. When configured, the time will be adjusted by changing to the configured offset when summer time starts and returning to the configured offset when summer time ends.

Use commands in the following context to configure the start day, end day, and offset of the summer.

```
configure system time dst-zone
```

If the time zone configured is listed in [Table 16: System-defined time zones and UTC offsets](#), the start and end command options and offset do not need to be configured with this command unless there is a need to override the system defaults. The command will return an error if the start and end dates and times are not available either in [Table 16: System-defined time zones and UTC offsets](#) or entered as optional command options in this command.

6.10.2.3 NTP

NTP is the Network Time Protocol defined in RFC 1305, *Network Time Protocol (Version 3) Specification, Implementation and Analysis* and RFC 5905, *Network Time Protocol Version 4: Protocol and Algorithms Specification*. It allows for the participating network nodes to keep time more accurately and more importantly they can maintain time in a more synchronized fashion between all participating network nodes.

SR OS uses an NTP process based on a reference build provided by the Network Time Foundation. Nokia strongly recommends that the users review RFC 8633, *Network Time Protocol Best Current Practices*, when they plan to use NTP with the router. The RFC section "Using Enough Time Sources" indicates that using only two time sources (NTP servers) can introduce instability if they provide conflicting information. To maintain accurate time, Nokia recommends configuring three or more NTP servers.

NTP uses stratum levels to define the number of hops from a reference clock. The reference clock is considered to be a stratum-0 device that is assumed to be accurate with little or no delay. Stratum-0 servers cannot be used in a network. However, they can be directly connected to devices that operate as stratum-1 servers. A stratum-1 server is an NTP server with a directly-connected device that provides Coordinated Universal Time (UTC), such as a GPS or atomic clock.

The higher stratum levels are separated from the stratum-1 server over a network path, therefore, a stratum-2 server receives its time over a network link from a stratum-1 server. A stratum-3 server receives its time over a network link from a stratum-2 server.

SR OS routers normally operate as a stratum-2 or higher device. The router relies on an external stratum-1 server to source accurate time into the network. However, SR OS also allows for the use of the local PTP recovered time to be sourced into NTP. In this latter case, the local PTP source appears as a stratum-0 server and SR OS advertises itself as a stratum-1 server. Activation of the PTP source into NTP may impact the network NTP topology because the SR OS router is promoted to stratum-1.

SR OS router runs a single NTP clock which then operates NTP message exchanges with external NTP clocks. Exchanges can be made with external NTP clients, servers, and peers. These exchanges can be through the base, management, or VPRN routing instances.

NTP operates associations between clocks as either client or server, symmetric active and symmetric passive, or broadcast modes. These modes of operation are applied according to which elements are configured on the router. To run server mode, the operator must enable NTP server mode for the base and each needed VPRN routing instance. To run client mode, the operator must configure external servers. If both the local router and remote router are configured with each other as peers, then the router operates in symmetric active mode. If only one side of the association has peering configured, then the modes are symmetric passive. To operate using broadcast mode, interfaces must be configured to transmit as broadcast servers or receive as broadcast clients.

NTP server operation for both unicast and broadcast communication within a VPRN is configured within the VPRN (see "NTP Within a VPRN Service" in the *7705 SAR Gen 2 Layer 3 Services Guide: IES and VPRN*).



Note: NTP provides lightweight synchronization across a network for alignment of system time for logging purposes. NTP does not provide the high accuracy time needed for the on-air applications of the mobile base stations.

The following NTP elements are supported:

- **server mode**

In this mode, the node advertises the ability to act as a clock source for other network elements. The node, by default, transmits NTP packets in NTP version 4 mode.

- **authentication keys**

Authentication keys implement increased security support in carrier and other networks. Both DES and MD5 authentication are supported, as well as multiple keys.

- **operation in symmetric active mode**

This capability requires that NTP be synchronized with a specific node that is considered more trustworthy or accurate than other nodes carrying NTP in the system. This mode requires that a specific peer is set.

- **server and peer addressing using IPv6**

Both external servers and external peers may be defined using IPv6 or IPv4 addresses. Other features (such as multicast, broadcast) use IPv4 addressing only.

- **broadcast or multicast modes**

When operating in these modes, the node receives or sends using either a multicast (default 224.0.1.1) or a broadcast address. Multicast is supported only on the CPM MGMT port.

- **alert when NTP server is not available**

When none of the configured servers are reachable on the node, the system reverts to manual timekeeping and issues a critical alarm. When a server becomes available, a trap is issued indicating that standard operation has resumed.

- **NTP and SNTP**

If both NTP and SNTP are enabled on the node, then SNTP transitions to an operationally down state. If NTP is removed from the configuration or shut down, then SNTP resumes an operationally up state.

- **gradual clock adjustment**

As several applications (such as Service Assurance Agent (SAA)) can use the clock, and if determined that a major (128 ms or more) adjustment needs to be performed, the adjustment is performed by programmatically stepping the clock. If a minor (less than 128 ms) adjustment must be performed, then the adjustment is performed by either speeding up or slowing down the clock.

- To avoid the generation of too many events/trap the NTP module rates limit the generation of events/traps to three per second. At that point a single trap is generated that indicates that event/trap squashing is taking place.

6.10.2.3.1 Authentication-check

NTP supports an authentication mechanism to provide some security and access control to servers and clients. The authentication check feature provides the option to skip the rejection of NTP PDUs that do not match the authentication key or authentication type requirements.

The default behavior when authentication is configured is to reject all NTP PDUs that have a mismatch in either the authentication key ID, type, or key.

When authentication check is configured, NTP PDUs are authenticated on receipt. However, mismatches cause a counter to be increased, one counter for key ID, one for type, and one for key value mismatches.

Use commands in the following context to enable authentication check.

```
configure system time ntp authentication-check
```

Example: MD-CLI

```
[ex:/configure system time ntp]
A:admin@node-2# info
    admin-state enable
    authentication-check true
```

Example: classic CLI

```
A:node-2>config>system>time>ntp# info
-----
                authentication-check
                no shutdown
-----
```

6.10.2.3.2 Authentication-key

The **authentication-key** command configures an authentication key ID, key type, and key used to authenticate NTP PDUs sent to and received from other network elements participating in the NTP. For

authentication to work, the authentication key ID, authentication type, and authentication key value must match.

Example: MD-CLI

```
[ex:/configure system time ntp]
A:admin@node-2# info
    admin-state enable
    authentication-key 1 {
        key "0AwgNULbZgI hash2"
        type des
    }
```

Example: classic CLI

```
A:node-2>config>system>time>ntp# info
-----
                shutdown
                authentication-key 1 key "0AwgNULbZgI" hash2 type des
-----
```

6.10.2.3.3 Broadcast

The **broadcast** command is used to transmit broadcast packets on a given interface. Interfaces in the base routing context or the management interface may be specified. Due the relative ease of spoofing of broadcast messages, it is strongly recommended to use authentication with broadcast mode. The messages are transmitted using a destination address that is the NTP Broadcast address. The following example enables NTP and configures the broadcast interface.

Example: MD-CLI

```
[ex:/configure system time ntp]
A:admin@node-2# info
    admin-state enable
    broadcast "Base" interface-name "int11" {
        version 4
        ttl 127
    }
```

Example: classic CLI

```
A:node-2>config>system>time>ntp# info
-----
                broadcast interface int11 version 4 ttl 127
                no shutdown
-----
```

6.10.2.3.4 Broadcastclient

The **broadcastclient** command enables listening to NTP broadcast messages on the specified interface. Interfaces in the base routing context or the management interface may be specified. Due the relative ease of spoofing of broadcast messages, it is strongly recommended to use authentication with broadcast mode. The messages must have a destination address of the NTP Broadcast address. The following example enables NTP and configures the broadcast client.

Example: MD-CLI

```
[ex:/configure system time ntp]
A:admin@node-2# info
    admin-state enable
    broadcast-client "Base" interface-name "int11" {
    }
```

Example: classic CLI

```
A:node-2>config>system>time>ntp# info
-----
                broadcastclient interface int11
                no shutdown
-----
```

6.10.2.3.5 Multicast

When configuring NTP the node can be configured to transmit or receive multicast packets on the CPM MGMT port. Broadcast and multicast messages can easily be spoofed; therefore, authentication is strongly recommended. Multicast is used to configure the transmission of NTP multicast messages. When transmitting multicast NTP messages the default address of 224.0.1.1 is used. The following example enables NTP and multicast.

Example: MD-CLI

```
[ex:/configure system time ntp]
A:admin@node-2# info
    admin-state enable
    multicast {
    }
```

Example: classic CLI

```
A:node-2>config>system>time>ntp# info
-----
                multicast
                no shutdown
-----
```

6.10.2.3.6 Multicastclient

The **multicastclient** command is used to configure an address to receive multicast NTP messages on the CPM MGMT port. Broadcast and multicast messages can easily be spoofed, therefore, authentication is strongly recommended. If multicastclient is not configured, all NTP multicast traffic is ignored. The following example enables NTP and configures the address to receive multicast NTP messages.

Example: MD-CLI

```
[ex:/configure system time ntp]
A:admin@node-2# info
    admin-state enable
    multicast-client {
        authenticate true
    }
```

```
}
```

Example: classic CLI

```
A:node-2>config>system>time>ntp# info
-----
          multicastclient
          no shutdown
-----
```

6.10.2.3.7 NTP-server

The **ntp-server** command configures the node to assume the role of an NTP server. Unless the **server** command is used, this node will function as an NTP client only and will not distribute the time to downstream network elements. If the **authenticate** command option is specified, the NTP server requires client packets to be authenticated.

The following is an example of a configuration output of NTP enabled with the **ntp-server** command configured.

Example: MD-CLI

```
[ex:/configure system time ntp]
A:admin@node-2# info
  admin-state enable
  ntp-server {
    authenticate true
  }
```

Example: classic CLI

```
A:node-2>config>system>time>ntp# info
-----
          no shutdown
          ntp-server
-----
```

6.10.2.3.8 Peer

Configuration of an NTP peer configures symmetric active mode for the configured peer. Although any system can be configured to peer with any other NTP node, Nokia recommends configuring authentication and to configure known time servers as their peers. Administratively disable this command to remove the configured peer.

Use commands in the following context to configure symmetric active mode.

```
configure system time ntp peer
```

Example: MD-CLI

```
[ex:/configure system time ntp]
A:admin@node-2# info
  admin-state enable
  peer 192.168.1.1 router-instance "Base" {
```

```
key-id 1
}
```

Example: classic CLI

```
A:node-2>config>system>time>ntp# info
-----
no shutdown
peer 192.168.1.1 key-id 1
-----
```

6.10.2.3.9 Server

The **server** command is used when the node should operate in client mode with the NTP server specified in the address field. Up to ten NTP servers can be configured. The following example enables NTP and configures the server.

Example: MD-CLI

```
[ex:/configure system time ntp]
A:admin@node-2# info
admin-state enable
server 192.168.1.1 router-instance "Base" {
key-id 1
}
```

Example: classic CLI

```
A:node-2>config>system>time>ntp# info
-----
no shutdown
server 192.168.1.1 key 1
-----
```

6.10.2.4 SNTP

SNTP is a compact, client-only version of NTP. SNTP can only receive the time from SNTP/NTP servers; it cannot be used to provide time services to other systems. SNTP can be configured in either broadcast or unicast client mode.

SNTP time elements include the [Broadcast-client](#) and [Server-address](#).

Use the commands in the following context to configure the SNTP.

```
configure system time sntp
```

6.10.2.4.1 Broadcast-client

Use the following command to enable listening at the global device level to SNTP broadcast messages on interfaces with broadcast client configured:

- **MD-CLI**

```
configure system time sntp sntp-state broadcast
```

- **classic CLI**

```
configure system time sntp broadcast-client
```

The following example shows SNTP enabled with the broadcast client.

Example: MD-CLI

```
[ex:/configure system time sntp]
A:admin@node-2# info
    admin-state enable
    sntp-state broadcast
```

Example: classic CLI

```
A:node-2>config>system>time>sntp# info
-----
                broadcast-client
                no shutdown
-----
```

6.10.2.4.2 Server-address

Use the following command to configure an SNTP server for SNTP unicast client mode:

- **MD-CLI**

```
configure system time sntp server
```

- **classic CLI**

```
configure system time sntp server-address
```

The following is an example of a configuration output of SNTP enabled with the **server-address** command configured.

Example: MD-CLI

```
[ex:/configure system time sntp]
A:admin@node-2# info
    admin-state enable
    server 10.10.0.94 {
        version 1
        prefer true
        interval 100
    }
```

Example: classic CLI

```
A:node-2>config>system>time>sntp# info
-----
                server-address 10.10.0.94 version 1 preferred interval 100
-----
```

```
no shutdown
-----
```

6.10.2.5 CRON

The CRON feature supports periodic and date and time-based scheduling in SR OS. CRON can be used, for example, to schedule Service Assurance Agent (SAA) functions. CRON functionality includes the ability to specify scripts that need to be run, when they are scheduled, including one-time only functionality (one-shot), interval and calendar functions. Scheduled reboots, peer turn ups, service assurance agent tests and more can all be scheduled with CRON, as well as OAM events, such as connectivity checks, or troubleshooting runs.

CRON supports the schedule element. The schedule function configures the type of schedule to run, including one-time only (one-shot), periodic, or calendar-based runs. All runs are determined by month, day of month or weekday, hour, minute, and interval (seconds).

6.10.2.5.1 Schedule

The **schedule** command configures the type of schedule to run, including one-time only (oneshot), periodic, or calendar-based runs. All runs are determined by month, day of the month or weekday, hour, minute, and interval (seconds). If the **end-time** and **interval** command options are both configured, whichever condition is reached first is applied.

The following is an example of a configuration output that schedules a script named "test" to run every 15 minutes on the 17th of each month and every Friday until noon on July 17, 2007.

Example: MD-CLI

```
[ex:/configure system cron]
A:admin@node-2# info
  schedule "test" owner "TiMOS CLI" {
    day-of-month [17]
    minute [0 15 30 45]
    weekday [friday]
    end-time {
      date-and-time 2007-07-17T12:00:00.0+00:00
    }
  }
```

Example: classic CLI

```
*A:node-2>config>system>cron# info
-----
  schedule "test"
    shutdown
    day-of-month 17
    minute 0 15 30 45
    weekday friday
    end-time 2007/07/17 12:00
  exit
-----
```

6.10.3 ANCP enhancements

Persistency is available for subscriber ANCP attributes and is stored on the on-board compact flash card. ANCP data stays persistent during an ISSU as well as node reboots. During recovery, ANCP attributes are first restored fully from the persistence file, and incoming ANCP sessions are temporarily on hold. Afterwards, new ANCP data can overwrite any existing values. This new data is then stored into the compact flash in preparation for the next event.

6.10.4 Configuring backup copies

The **config-backup** command allows you to specify the maximum number of backup versions of configuration and index files kept in the primary location.

For example, assume the maximum number of backup versions is set to 5 and the configuration file is called `xyz.cfg`. When the configuration is saved, the file `xyz.cfg` is saved with a `.1` extension. Each subsequent **config-backup** command increments the numeric extension until the maximum count is reached. The oldest file (5) is deleted as more recent files are saved.

- `xyz.cfg`
- `xyz.cfg.1`
- `xyz.cfg.2`
- `xyz.cfg.3`
- `xyz.cfg.4`
- `xyz.cfg.5`
- `xyz.ndx`

Each persistent index file is updated at the same time as the associated configuration file. When the index file is updated, the save is performed to `xyz.cfg` and the index file is created as `xyz.ndx`. Synchronization between the active and standby SF/CPMSF/CPM is performed for all configurations and their associated persistent index files.

Use the following commands to specify the maximum number of backup versions of the configuration and index files kept in the primary location:

- **MD-CLI**

```
configure system management-interface configuration-save configuration-backups
```

- **classic CLI**

```
configure system config-backup
```

6.11 Configuring power supply

The 7705 SAR-1 supports component redundancy for the power supply feeds. The use of up to two power supply feeds for 1+1 power redundancy is supported. A power feed can fail without impacting traffic when a redundant power feed configuration is in use.

The following output shows the chassis power supply information for the 7705 SAR-1.

Example

```
A:node-2 show chassis power-supply
=====
Chassis 1 Detail
=====
Power Supply Information
  Number of power supplies      : 2

  Power supply number          : 1
    Power supply type          : none
    Status                     : failed

  Power supply number          : 2
    Power supply type          : none
    Status                     : failed
=====
```

6.12 Configuring multichassis redundancy for LAG

When configuring the associated LAG ID, the LAG must be in access mode and LACP must be enabled. The following example configures multichassis redundancy features.

Example: MD-CLI

```
[ex:/configure redundancy multi-chassis]
A:admin@node-2# info
  peer 10.10.10.2 {
    admin-state enable
    description "Mc-Lag peer 10.10.10.2"
    mc-lag {
      admin-state enable
      lag "lag-1" {
        lacp-key 32666
        system-id 00:00:00:33:33:33
        system-priority 32888
      }
    }
  }
}
```

Example: classic CLI

```
A:node-2>config>redundancy>multi-chassis# info
-----
  peer 10.10.10.2 create
    description "Mc-Lag peer 10.10.10.2"
    mc-lag
      no shutdown
    exit
  no shutdown
exit
-----
```


6.13 Post-boot configuration extension files

Two post-boot configuration extension files are supported and are triggered when either a successful or failed boot configuration file is processed. The commands specify URLs for the CLI scripts to be run following the completion of the startup configuration. A URL must be specified or no action is taken. The commands are persistent between router reboots and are included in the configuration saves.

Use the following commands to specify the CLI scripts that are run following the completion of the boot-up configuration.

```
configure system boot-bad-exec
configure system boot-good-exec
```

6.13.1 Show command output and console messages

Use the following command to show the current value of the bad and good exec URLs and indicate whether a post-boot configuration extension file was executed when the system was booted.

```
show system information
```

If an extension file was executed, the command also indicates whether it completed successfully.

The following example shows the show output for the 7705 SAR-1.

Output example: Show system information output for the 7705 SAR-1

```
=====
System Information
=====
System Name       : node-2
System Type      : 7705 SAR-1
...
BOF Source       : cf3:
Image Source     : primary
Config Source    : N/A
Last Booted Config File: N/A
Last Boot Cfg Version : N/A
Last Boot Config Header: N/A
Last Boot Index Version: N/A
Last Boot Index Header : N/A
Last Saved Config  : ftp://test:test@192.168.xx.xxx/./images/dut-bg.cfg
Time Last Saved   : 2025/03/27 20:12:15
Changes Since Last Save: No
User Last Modified : admin
Max Cfg/BOF Backup Rev : 5
Cfg-OK Script     : ftp://test:test@192.168.xx.xxx/./images/env.cfg
Cfg-OK Script Status : success
Cfg-Fail Script   : N/A
Cfg-Fail Script Status : not used
...
=====
```

When executing a post-boot configuration extension file, status messages are displayed on the console before the "Login" prompt.

The following example shows a failed bootup configuration that caused a boot-bad-exec file containing another error to be executed.

Example: Failed start-up configuration error message

```
Attempting to exec configuration file:
'ftp://test:test@192.168.xx.xxx/./12.cfg' ...
System Configuration
Log Configuration
MAJOR: CLI #1009 An error occurred while processing a CLI command -
File ftp://test:test@192.168.xx.xxx/./12.cfg, Line 195: Command "log" failed.
CRITICAL: CLI #1002 An error occurred while processing the configuration file.
The system configuration is missing or incomplete.
MAJOR: CLI #1008 The SNMP daemon is disabled.
If desired, enable SNMP with the 'config>system>snmp no shutdown' command.
Attempting to exec configuration failure extension file:
'ftp://test:test@192.168.xx.xxx/./fail.cfg' ...
Config fail extension
Enabling SNMP daemon
MAJOR: CLI #1009 An error occurred while processing a CLI command -
File ftp://test:test@192.168.xx.xxx/./fail.cfg, Line 5: Command "abc log" failed.
TIMOS-B-x.0.Rx both/hops ALCATEL Copyright (c) 2000-2001 Alcatel-Lucent.
All rights reserved. All use subject to applicable license agreements.
Built on Thu Nov 207 19:19:11 PST 2008 by builder in /rel5x.0/b1/Rx/panos/main

Login:
```

6.14 Configuring system monitoring thresholds

This section provides information about configuring system monitoring thresholds.

6.14.1 Creating events

The **event** command controls the generation and notification of threshold crossing events configured with the **alarm** command. When a threshold crossing event is triggered, the **rmon event** configuration optionally specifies whether an entry in the RMON-MIB log table is created to record the occurrence of the event. It can also specify whether an SNMP notification (trap) be generated for the event. There are two notifications for threshold crossing events: a rising alarm and a falling alarm.

Creating an event entry in the RMON-MIB log table does not create a corresponding entry in the event logs. However, when the event is set to trap, the generation of a rising alarm or falling alarm notification creates an entry in the event logs and that is distributed to whatever log destinations are configured: console, session, memory, file, syslog, or SNMP trap destination. The logger message includes a rising or falling threshold crossing event indicator, the sample type (absolute or delta), the sampled value, the threshold value, the RMON alarm ID, the associated RMON event ID, and the sampled SNMP object identifier.

The **alarm** command configures an entry in the RMON-MIB alarm table. The **alarm** command controls the monitoring and triggering of threshold crossing events. For notification or logging of a threshold crossing event to occur, there must be at least one associated **rmon event** configured.

The agent periodically takes statistical sample values from the MIB variable specified for monitoring and compares them to thresholds that have been configured with the **alarm** command. The **alarm** command

configures the MIB variable to be monitored, the polling period (interval), sampling type (absolute or delta value), and rising and falling threshold parameters. If a sample has crossed a threshold value, the associated event is generated.

Preconfigured CLI threshold commands are available. Preconfigured commands hide some of the complexities of configuring RMON alarm and event commands and perform the same function. The preconfigured commands do not require the user to know the SNMP object identifier to be sampled. The preconfigured threshold configurations include memory warnings and alarms and compact flash usage warnings and alarms.

Example: MD-CLI

```
[ex:/configure system thresholds]
A:admin@node-2# info
  cflash-cap-warn-percent "cfl-B:" {
    rising-threshold 100
    falling-threshold 50
    interval 240
    startup-alarm either
  }
  kb-memory-use-alarm {
    rising-threshold 50000000
    falling-threshold 45999999
    interval 500
    startup-alarm either
  }
  rmon {
    event 5 {
      description "alarm testing"
      owner "Timos CLI"
    }
  }
}
```

Example: classic CLI

```
A:node-2>config>system>thresholds# info
-----
      rmon
        event 5 description "alarm testing" owner "Timos CLI"
      exit
      cflash-cap-warn cfl-B: rising-threshold 2000000 falling-threshold
1999900 interval 240 trap
        memory-use-alarm rising-threshold 50000000 falling-threshold
45999999 interval 500
-----
```

6.15 Configuring LLDP

Use the commands in the following context to configure LLDP.

```
configure system lldp
```

Example: LLDP default configuration (MD-CLI)

```
[ex:/configure system lldp]
A:admin@node-2# info detail
```

```
## apply-groups
## apply-groups-exclude
admin-state enable
tx-credit-max 5
message-fast-tx 1
message-fast-tx-init 4
tx-interval 30
tx-hold-multiplier 4
reinit-delay 2
notification-interval 5
```

Example: LLDP default configuration (classic CLI)

```
A:node-2>config>system>lldp# info detail
-----
no tx-interval
no tx-hold-multiplier
no reinit-delay
no notification-interval
no tx-credit-max
no message-fast-tx
no message-fast-tx-init
no shutdown
-----
```

Example: LLDP port configuration (MD-CLI)

```
[ex:/configure port 1/1/1 ethernet lldp]
A:admin@node-2# info
  dest-mac nearest-bridge {
    receive true
    transmit true
    tx-tlvs {
      port-desc true
      sys-cap true
    }
    tx-mgmt-address system {
      admin-state enable
    }
  }
```

Example: LLDP port configuration (classic CLI)

```
A:node-2>config>port>ethernet>lldp# info
-----
dest-mac nearest-bridge
  admin-status tx-rx
  tx-tlvs port-desc sys-cap
  tx-mgmt-address system
exit
-----
```

Example: Global system LLDP configuration (MD-CLI)

```
[ex:/configure system lldp]
A:admin@node-2# info
tx-interval 10
tx-hold-multiplier 2
reinit-delay 5
notification-interval 10
```

Example: Global system LLDP configuration (classic CLI)

```
A:node-2>config>system>lldp# info
```

```
-----  
tx-interval 10  
tx-hold-multiplier 2  
reinit-delay 5  
notification-interval 10  
-----
```

7 Zero touch provisioning

Traditional deployment of new nodes in a network is a multi-step process in which the user connects to the hardware to provision global and local configuration options. ZTP automatically configures the node by obtaining the required information from the network and automatically provisioning the node with minimal manual intervention and configuration. When nodes that support ZTP are installed in the rack, connected to the network, and powered on, the nodes are auto-provisioned.



Note: To support ZTP, make sure the new nodes are purchased with the **auto-boot** flag enabled in the factory-loaded BOF.

7.1 ZTP overview

ZTP is used to automatically install and provision new nodes in the field. For out-of-band management, the nodes can be installed and powered up with network connectivity on the management (Mgmt) port. For in-band management, the first two connectors on the first two slots can be used for ZTP.



Note: For breakout connectors, only the first breakout port on the first two connectors can be used for ZTP.

ZTP VLAN discovery is enabled by default.

After network connectivity is established, the ZTP process starts automatically. The node sends a DHCP discovery request to the DHCP server using a ZTP-capable port and the DHCP server returns an IPv4/IPv6 FTP or HTTP URL from which the provisioning information can be retrieved. The provisioning information is in a file called the provisioning file, which contains the URL of the image, config, and other files to be downloaded. After downloading these files and successfully provisioning, the node automatically reboots and comes back up in normal mode.

Secure ZTP (SZTP), which is an extension of ZTP, is also supported. See [SZTP](#) for information about SZTP.



Note: ZTP supports only TLS 1.2 for HTTPS.

7.1.1 Network requirements

ZTP requires the following network components:

- **DHCP server (IPv4 or IPv6)**

The DHCP server supports assignment of IP addresses through DHCP requests and offers.

- **file server**

The FTP or HTTP file server is used for staging and transfer of RPMs, configurations, images, and scripts.

- **DHCP relay**

A DHCP relay is required if the servers are across a Layer 3 network.

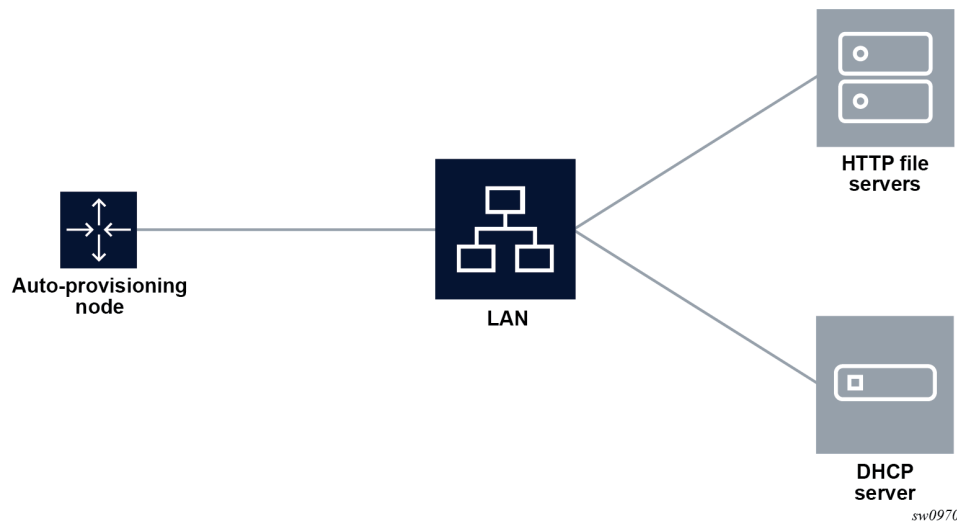
7.1.2 Network support

ZTP operates in the following network environments:

- **node, file servers, and DHCP server in the same subnet**

The following figure shows the scenario where all components are in a Layer 2 broadcast domain. There is no DHCP relay and all IP addresses are assigned from a single pool.

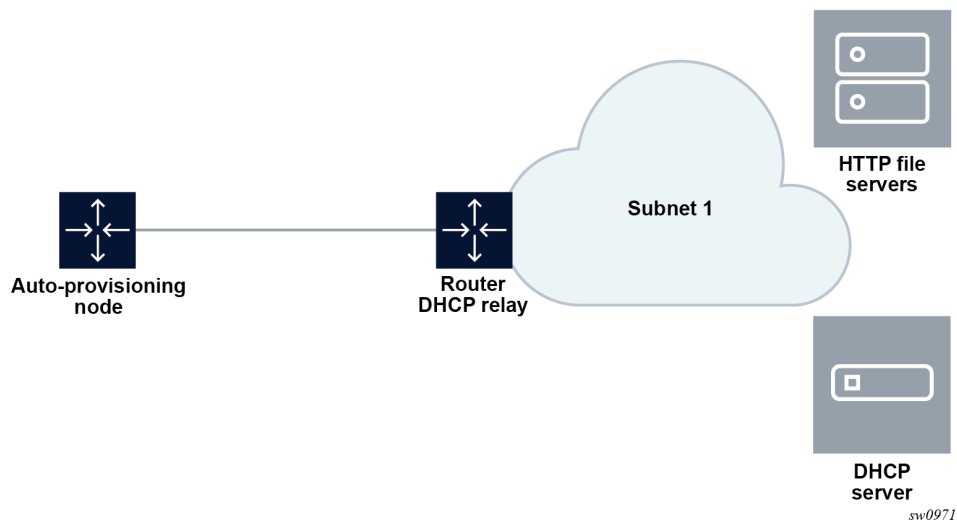
Figure 10: Auto-provisioning with all components in the same subnet



- **file servers and DHCP server in the same subnet, separate from the nodes**

The following figure shows the scenario where only the file servers and DHCP server are in the same subnet. The DHCP relay is used to fill option 82 as the gateway address. The gateway address is used to find the appropriate pool in the DHCP server to assign the correct subnet IP address to the system.

Figure 11: Auto-provisioning with only file and DHCP servers in the same subnet

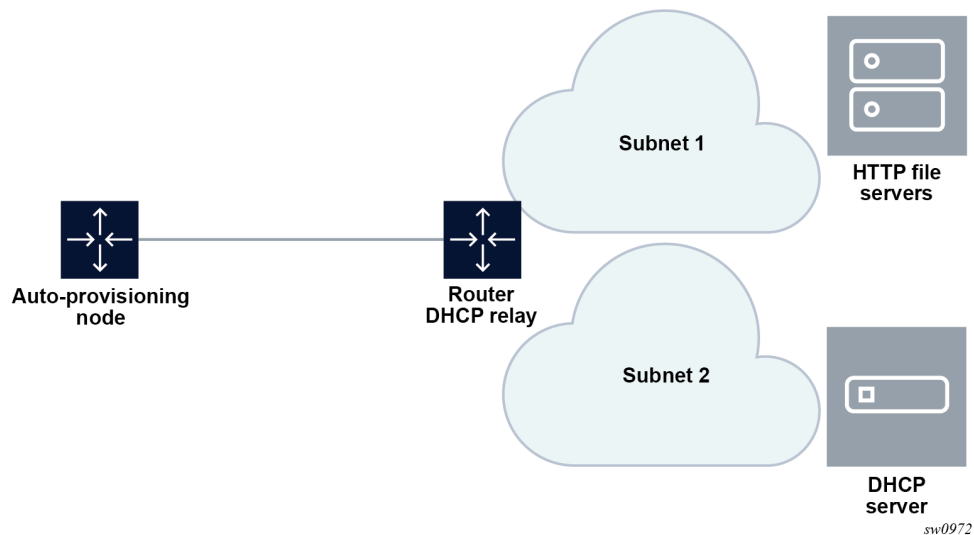


DHCP allows the Option 3 router to define the default gateway. If multiple addresses are provided using Option 3, the first address is used for the default gateway.

- **node, file servers, and DHCP server in different subnets**

The following figure shows the scenario where all components are in different subnets. The DHCP relay adds the option 82 gateway address to the DHCP request, and the DHCP server adds the option 3 with the gateway address of the file server.

Figure 12: Auto-provisioning with all components in different subnets



7.2 ZTP process overview

ZTP consists of the following processes:

- [Auto-boot process](#)
- [Auto-provisioning process](#)

7.2.1 Auto-boot process

In this process, the node discovers and provisions the chassis and installed cards.

1. The node is first connected to the network and powered on.
2. The out-of-band management port is checked for link connectivity. If a link is not found, the system checks the in-band management ports for a link.
3. The first two card or MDA slots are auto-provisioned based on the installed card types. See [ZTP overview](#) for information about the specific card or MDA slots that are used.
4. The auto-boot process switches control to the auto-provisioning process.

See [Auto-boot process details](#) for more information about the auto-boot process.

7.2.2 Auto-provisioning process

In this process, the node detects operational ports, attempts to discover its IP address, and downloads the relevant files for provisioning.

1. The node sends a DHCP discovery request to the DHCP server using the out-of-band management port. If DHCP discovery is unsuccessful, the node reattempts it using the in-band management ports.
2. After DHCP discovery is successful, the DHCP server returns an IPv4 or IPv6 FTP or HTTP URL of a file server from which the node can retrieve provisioning information.
3. The node downloads the provisioning information and performs the auto-provisioning according to the specifications in the files.
4. After the node is successfully provisioned, it automatically reboots and becomes operationally up.

See [Provisioning files](#) for more information about the auto-provisioning process.

The SR OS can also initiate the auto-provisioning process using a **tools** command.

7.3 DHCP support for ZTP

This section provides information about DHCP messages, DHCP clients, and DHCP servers that are supported by ZTP.

7.3.1 DHCP server offer options

Options 66, 67, and 43 are supported for indicating the location of the provisioning file. If both Options 66 and 67 are present in the DHCP offer, they take precedence over Option 43.

Option 66 contains the server URL or IP address, and Option 67 contains the URL of the provisioning file location.

Options 66 and 67 are meant for use by PXE TFTP, but are also used for HTTP and FTP. If an offer arrives with Options 66 and 67, Option 66 should resolve the server IP address and Option 67 should resolve the file location. Option 66 can be omitted by the provider, in which case Option 67 is used for both the server IP address and provider file URL. If an offer arrives with Option 67 only, it should resolve both the server IP address and file URL.

The auto-provisioning process distinguishes the host part of the URL and can resolve it using DHCP DNS.

7.3.1.1 Nokia-specific TLV

The Nokia-specific TLV is NOKIA-DCTOR-AUTOCONFIG. The location of the BOF for each system to use is configured in the optional **autoboot** file parameter, which is a standard Option 43 value initialized at the beginning of the process. The BOF location is sent in Option 43 as part of the DHCP offer and Ack messages from the DHCP server to the system. The system uses the location specified in Option 43 to initiate an FTP download of the BOF.

7.3.2 Supported DHCP client options for ZTP

The following table lists the supported DHCP client options for ZTP.

Table 17: Supported DHCP client options for ZTP

Options	DHCP IPv4 option	IPv4 comments	DHCP IPv6 option	IPv6 comments
Lease time	Option 51	Always infinite	—	—
Requested option list	Option 55	—	—	—
Client ID	Option 61	Default is chassis serial ID	Option 1 (DUID)	Type 2 — vendor-assigned unique ID (default with chassis serial ID) Type 3 — link-layer address
User class	Option 77	"platform;timos-release;ztp"	Option 15	"platform;timos-release;ztp"
Class ID	Option 60	"NOKIA: Fmt ChassisType Strings"	—	—

7.3.3 Supported DHCP server options for ZTP

The following table lists the supported DHCP server options for ZTP.

Table 18: Supported DHCP server options for ZTP

Options	DHCP IPv4 option	IPv4 comments	DHCP IPv6 option	IPv6 comments
Subnet mask	Option 1	—	—	—
Router	Option 3	Default gateway	—	—
DNS server	Option 6	DNS server	—	—
Lease time	Option 51	Must be infinite	—	—
Server address	Option 54	Identifies the DHCP server	—	—
Classless static route	Option 121	Used to install static routes	—	—
NTP server ¹	Option 42	—	Option 56	—
TFTP server name	Option 66	Server IP address	—	—
Bootfile name	Option 67	URL of the file This option can be used without Option 66, in which case it contains the server name and the URL	Option 59	Server name and URL of the file
Vendor-specific options (See Nokia-specific TLV)	Option 43	Nokia proprietary file location; can be used instead of Options 66 or 67, but Options 66 and 67 take precedence over Option 43	Option 17	Nokia proprietary file location; can be used instead of Option 59, but Option 59 takes precedence over Option 17

7.3.4 DHCP discovery and solicitation

IPv4 DHCP discovery and IPv6 DHCP solicitation are supported.

¹ When the node is running in ZTP mode, the date and time are set by NTP. This information is required for HTTPs certificate verification, and to record date and time stamps in events and logs.

IPv4 DHCP discovery messages and IPv6 DHCP solicitation messages are sent from out-of-band and in-band management ports with active links. The first valid DHCP offer for the address family that arrives on the node is used.

In the BOF, the **auto-boot** option can be configured to send out IPv4, IPv6, or both IPv4 and IPv6 DHCP requests.

7.3.4.1 DHCP discovery (IPv4 and IPv6)

This section describes DHCP discovery options.

7.3.4.1.1 DHCP discovery Options 61 and 77

SR OS supports both Option 61 (client ID) and Option 77 (user class) DHCP discovery options.

Option 61 is used for DHCP server pool selection. Option 61 provides the client ID; the serial ID of the chassis with a type of 0 is used by default. This option is configurable using commands in the **bof auto-boot** context.

Option 77 provides the user class, describing what the device is and other information, such as the OS version. This option is set automatically, but can be removed using the BOF configuration. For example, the user can delete **include-user-class** in the BOF auto-boot configuration to avoid sending Option 77.

For ZTP, the DHCP discovery message should be sent with Option 77; the following information is automatically configured:

```
platform;timos-release;ztp
```

For auto-provisioning, Option 77 should use the following information:

```
platform;timos-release;AP
```

7.3.4.1.2 DHCP discovery Option 1 DUID (IPv6)

By default, the node uses RFC 3315 DUID Type 2 vendor-assigned unique IDs. The value for *enterprise-id* is 6527 and the identifier is the chassis serial number.



Note: The system uses the chassis serial number for ZTP pool selection and auto-provisioning.

The option to use Type 3 is configured in the BOF. For MAC, the chassis MAC address is configured in a string format.

Type 1 is not supported.

7.3.4.2 DHCP solicitation (IPv6)

Unlike IPv4 DHCP offers, which contain the prefix and default route, IPv6 DHCP offers only contain the IP address assignment. The IPv6 route advertisement (RA) provides the default router and the prefix is set to /128 for the IP address supplied by the DHCP server.

For further information about RA support, see [IPv6 DHCP/RA details](#). For further information about DHCP server offers, see [DHCP server offer options](#).

7.3.5 IPv4 and IPv6 DHCP support

The ZTP process supports the use of IPv4 and IPv6 DHCP clients to obtain the provisioning file.

For ZTP processes, the node transmits both IPv4 and IPv6 discovery and solicitation messages. If offers arrive from both IPv4 and IPv6 servers, both offers are cached and the first offer received is processed. If the first offer does not fulfill the ZTP requirements and is rejected, the second offer is processed and accepted or rejected. If both offers received on an interface are rejected, ZTP goes to the next interface.

The provisioning file only allows file transfer in the address family of the DHCP offer that is used. If the offer is IPv4, the provisioning files are downloaded using IPv4. If the offer is IPv6, the provisioning files are downloaded using IPv6.

7.3.5.1 IPv4 route installation details

Option 3 (default route) and Option 121 (classless static route) are supported for IPv4 DHCP.

For identical routes with different next hops, only the first route is installed and the second route is kept as a backup route. ECMP is not supported.

There is no route limit for Option 121.

7.3.5.2 IPv6 DHCP/RA details

IPv6 DHCP offers do not contain an IP prefix. The IPv6 prefix is usually obtained from the IPv6 RA arriving from the upstream router. Because the SR OS router is the host for the ZTP process, the system assigns a /128 prefix to the IPv6 address obtained from the DHCP offer.

SR OS supports the use of an IPv6 RA to install IPv6 default and static routes from upstream routers. The system installs all the routes advertised using the RA. If the same route has been advertised from multiple upstream routers (next hops), the system installs the route with the highest preference. SR OS does not support ECMP if the same route is advertised from multiple next hops by multiple RAs.

In accordance with RFC 4861 recommendations, SR OS ensures that all RAs are obtained before the auto-provisioning process is started for IPv6. RFC 4861 recommends that the host (in this case, the SR OS router) send a minimum of three route solicitations to increase the likelihood of at least one route solicitation being received by the upstream routers. Each route solicitation is followed by a 4-second timeout, so the third route solicitation is sent 8 seconds after the first. The upstream routers must respond within 0.5 seconds. As a result, the SR OS router receives all RAs and routes within 8.5 seconds of the first route solicitation, and waits a maximum of 9 seconds to receive all RAs; ZTP always waits 20 seconds to receive all RAs, however, only the first RA received is used.

7.3.5.3 ZTP and DHCP timeouts

The ZTP timeout is user-configurable with a default value of 30 minutes. See [Options and option modification](#) and [Configuring the ZTP timeout in the provisioning file](#) for more information. After each ZTP timeout, the node reboots and reattempts the ZTP process. If the ZTP timeout interval expires while the node is executing a DHCP offer or downloading files, the node does not reboot. The DHCP offer is executed until it succeeds or fails, at which point the node reboots. If the offer is successful, the node comes up in normal operation mode.

The DHCP timeout interval is 20 seconds. If a DHCP offer is not received within the DHCP timeout interval, the auto-provisioning process is reattempted using the next valid interface.

7.4 ZTP procedure details

This section describes ZTP procedures including node bootup, BOF, auto-provisioning, logs, and events.

7.4.1 Node bootup

After the node is powered up, the BOF is examined for the **auto-boot** flag status. If the **auto-boot** flag is set in the `bof.cfg` file, the node goes into ZTP mode. If the **auto-boot** flag is not set in the `bof.cfg` file, the node continues booting normally.

If it is in ZTP mode, the node provisions all hardware necessary for the ZTP process. This includes the fabric, the first two card slots, and the MDAs for the first two card slots. The node then checks for links on the management (Mgmt) port and valid Ethernet ports.



Note: A `bof.cfg` file with the **auto-boot** flag enabled can be shipped as an orderable part with the applicable software license. The **auto-boot** flag can also be set using the **bof auto-boot** command.

For more information about the BOF, see [BOF](#).

7.4.1.1 Reinitiating ZTP during normal node bootup

ZTP can be reinitiated any time by setting the **auto-boot** flag and configuring the flag options in the BOF. After the auto-boot flag is set, any reboot forces the node into ZTP mode, including DHCP discovery, and downloading and reprocessing the provisioning file. The old BOF is kept in the storage medium until the ZTP process is successful, then the old BOF information is overwritten. If an unsuccessful ZTP process is interrupted and the **auto-boot** flag is removed, the node boots using the old BOF.

7.4.2 BOF

Two versions of each supported 7705 SAR Gen 2 platform software license are currently available: one for non-ZTP bootup, and one for ZTP bootup. Software packages for ZTP bootup contain a `bof.cfg` file with the **auto-boot** flag set, which causes the node to automatically boot up in ZTP mode and execute ZTP processes.

The **auto-boot** flag contains the following information:

- **client ID**

The client ID is sent to the DHCP server to identify the chassis or node and to find a pool for the DHCP offer. If no client ID is configured, the chassis serial number is sent.

This option is used for both IPv4 client ID and IPv6 DUID Type 2.

- **port (port:vlan)**

The port is used to send DHCP discovery; the port number must be configured manually in the BOF.

For more information about the BOF, see [System initialization and boot options](#).

7.4.2.1 Compact flash support

The BOF itself does not support loading from the network using HTTP or HTTPS.

7.4.3 Auto-boot process details

This section describes the ZTP auto-boot process.

7.4.3.1 Options and option modification

By default, the auto-boot process scans all ZTP-enabled ports to find a port with an operational link. The scanned ports include:

- out-of-band management port (Mgmt port)
- Ethernet ports on the first two card or MDA slots (used for in-band management)



Note: For breakout connectors, only the first breakout port in the connector can be used for ZTP.

ZTP attempts to discover the node IP via DHCP and identifies the node using DHCP client ID Option 61 (IPv4) or Option 1 (IPv6). The client ID uses the chassis serial number by default. The chassis serial number is visible on the shipping box of the chassis.

[Table 17: Supported DHCP client options for ZTP](#) lists the default DHCP client options for ZTP. Some client options can be manually configured in the BOF using the **bof auto-boot** command.

The optional **auto-boot** configuration options are as follows:

- **management port**
Specify that ZTP should only be performed using the out-of-band management port (Mgmt port).
- **in-band VLAN**
Specify ZTP should only be performed using Ethernet ports on the first two card or MDA slots. The VLAN ID can be used to specify an in-band VLAN to use for the auto-boot process.
- **IPv4, IPv6**
Specify that IPv4 discovery, IPv6 discovery, or both, should be performed. If both are specified, the system dual-stacks.
- **client identifier**
Identify the node to the DHCP server and find a pool for DHCP offers. This information is sent using Option 61 (IPv4) or Option 1 (IPv6). If the **client-identifier** options are not configured, the chassis serial number is sent by default. This option is used for both IPv4 client ID and IPv6 DUID Type 2.
- **include user class**
Specify to include Option 77.
- **timeout**

Specify in minutes the timeout for the ZTP process to be executed successfully before the node is rebooted and ZTP is retried because of an unsuccessful ZTP completion. The default ZTP timeout is 30 minutes.

See [Configuring the ZTP timeout in the provisioning file](#) for information about how to configure the ZTP timeout in a ZTP provisioning file.

The **auto-boot** options can be modified using the **bof auto-boot** command, or by interrupting the bootup process and manually modifying the `bof.cfg` file.



Caution: Manually modifying the `bof.cfg` file is not recommended. When modifying **auto-boot** options using CLI, all required options must be explicitly configured because the default cases are no longer used. When modifying the `bof.cfg` file manually, the format must be correct.

7.4.3.2 CLI access

The auto-boot process is executed in the background and does not block the CLI. The user can enter CLI commands while the auto-boot process runs in the background. A warning message is displayed to notify the user that the auto-boot process is being executed. Any configurations performed using the CLI may be lost when the node reboots following successful auto-boot and auto-provisioning processes. After the node has finished booting and if the **auto-boot** flag is set in the BOF, the node displays the login prompt.

The user can access the CLI using a console and can change and save the BOF configuration; as such, the user can remove or modify the **auto-boot** option in the BOF.

7.4.3.3 Interrupting auto-boot

The auto-boot process can be interrupted using the **tools auto-boot terminate** command. After the auto-boot process is terminated, use the **bof auto-boot** command to modify the **auto-boot** flag.



Note: The **auto-boot** flag can also be modified without interrupting the auto-boot process.

7.4.4 Auto-provisioning process

In this process, the node detects operational ports, attempts to discover its IP address, and downloads the relevant files for provisioning.

1. The node sends a DHCP discovery request to the DHCP server using the out-of-band management port. If DHCP discovery is unsuccessful, the node reattempts it using the in-band management ports.
2. After DHCP discovery is successful, the DHCP server returns an IPv4 or IPv6 FTP or HTTP URL of a file server from which the node can retrieve provisioning information.
3. The node downloads the provisioning information and performs the auto-provisioning according to the specifications in the files.
4. After the node is successfully provisioned, it automatically reboots and becomes operationally up.

See [Provisioning files](#) for more information about the auto-provisioning process.

The SR OS can also initiate the auto-provisioning process using a **tools** command.

7.4.4.1 VLAN discovery

The node can perform VLAN discovery if it is shipped in ZTP mode. VLAN discovery is supported only for the in-band management port. It is not supported for the out-of-band management ports.

After the node is installed and powered up:

1. ZTP is attempted on the null (untagged) port first, including the out-of-band management port, and then on all in-band management ports with operational links.
 - a. SR OS scans each port with an operational link and sends IPv4 DHCP discovery messages.
 - b. SR OS waits for the DHCP offer within the DHCP timeout.
2. The first VLAN with a valid offer that includes the IPv4 DHCP Options 66 and 67, or Option 67 or 43, or IPv6 DHCP Option 59 or 17 is selected as the working VLAN and the ZTP process is executed on this VLAN.



Note: If there is no offer or the offer does not have the relevant or correct options, SR OS floods the network with DHCP discovery messages on all remaining non-reserved VLANs (1 to 4094).

3. When a VLAN is discovered, the ZTP process is executed on the respective VLAN as described in the following sections.



Note: If there is no offer on any VLAN or the offer does not have the relevant or correct options, the node starts over from step 1.

7.4.4.1.1 VLAN discovery option

By default, the auto-boot flag in the `bof . cfg` file has the VLAN discovery option enabled. The option can be disabled manually in the `bof . cfg` file or implicitly from the CLI BOF menu, using the command **bof auto-boot inband**. When the VLAN discovery option is disabled, the node executes the ZTP process using the untagged method only.

7.4.4.2 Auto-provisioning procedure

After the node enters ZTP mode, the auto-discovery process is executed to provision the necessary hardware for node discovery.

The following are the operational steps of the auto-discovery process.

1. DHCP is used to discover the IP address of the node.
2. Options 66 and 67, or Option 43 is used to find and download the provisioning file.

The provisioning file includes the location of necessary files, such as configuration information, system image, and licenses, along with the DNS needed to resolve these location URLs. The file also includes BOF information required to boot the node into operational mode.

3. The provisioning file is executed to download the named files to the node.
4. After all files are successfully downloaded, the node is rebooted and the **auto-boot** flag is cleared from the BOF.

After the node reboots, it comes up in normal operational mode.

The node can be put back into ZTP mode by editing the BOF to include the **auto-boot** flag and saving the BOF. Doing this causes the node to enter ZTP mode after it is rebooted.

Use one of the following methods to run the auto-provisioning process.

- **automatic execution**

The auto-boot process automatically executes the auto-provisioning process if the **auto-boot** flag is set in the BOF.

- **manual execution**

The auto-provisioning process can be executed manually using the following command.

```
tools perform system auto-node-provisioning
```

If the auto-provisioning process is executed manually, only interfaces without IP addresses are considered part of the discovery mechanism. Additionally, while the process is running, it attempts to discover DHCP servers using all card or MDA slots and ports with Layer 3 interfaces that do not have IP addresses.



Note: Using the following command while the auto-boot process is running is not allowed.

```
tools perform system auto-node-provisioning
```

7.4.4.3 Out-of-band management versus in-band management

The auto-provisioning process can use the out-of-band management port (Mgmt port), or in-band management on Ethernet ports.

The node attempts the auto-provisioning process using any port with an operational link, starting with the out-of-band management port. If the node cannot be discovered using the out-of-band management port, either because the port is down or the port is not receiving a DHCP offer from the DHCP server, the process is reattempted using Ethernet ports. If the node cannot be discovered using the Ethernet ports, the process is reattempted using the out-of-band management port and the cycle repeats.

The following operational guidelines apply to in-band and out-of-band management ports:

- Out-of-band management and in-band management support untagged frames.
- Out-of-band management does not support dot1q (VLAN tags).
- In-band management supports dot1q interfaces if the VLAN is correctly configured in the BOF.
- In-band ports support VLAN discovery for IPv4 by default, if not disabled in the BOF.

If out-of-band management is used, no card or MDA provisioning is necessary and the auto-provisioning process executes as soon as an active link is detected on the Mgmt port.

To use out-of-band management exclusively, use the following command:

```
bof auto-boot management-port
```

To use in-band management exclusively, use the following command:

```
bof auto-boot inband vlan
```

7.4.4.3.1 Supported in-band management ports

See [ZTP overview](#) for information about which ports support in-band management for ZTP.

7.4.5 Provisioning files

Provisioning files are created by the operator based on requirements and the locations of the necessary files. A provisioning file contains the locations and URLs of critical files such as the system image, configuration files, and necessary licenses, and can also contain DNS server information used to resolve these locations.

A provisioning file consists of two main parts:

- **locations of files**

Contains locations of the following file types:

- system image
- configuration files
- licenses

These items can be downloaded using HTTP, HTTPS, or FTP; DNS server information can also be included.



Note: If classic configuration mode is required when booting with ZTP, configuration files must have **exit all** as the first executable line.

- **BOF information**

Contains BOF information to be loaded on the node after the ZTP processes are completed; the BOF section of the file must be formatted correctly.



Caution: Ensure that the **auto-boot** flag is not set on the BOF that will be downloaded by the auto-provisioning process; failure to do so will cause the node to go back into ZTP mode after it reboots.

The provisioning file can be executed using the following command to re-download the named files:

```
tools perform system auto-node-provisioning file
```

7.4.5.1 Provisioning file download

The provisioning file location is discovered using DHCP offer Options 66 and 67 or Option 43, and is downloaded using HTTP or FTP.

The provisioning file URL can be resolved using DNS, in which case the IP addresses for up to three DNS servers should be present in the DHCP offer using Option 6 (IPv4). The DHCP DNS is only used for resolving the provisioning file URL, and not for resolving the URLs of the files named within the provisioning file.

ZTP does not support Option 15 domain names; the URL of the provisioning file should be in "*host/domain*" format, or a simple IP address should be used.

7.4.5.2 Provisioning file resolution using DNS

If the downloaded provisioning file includes a DNS IP in the DNS section of the file, the URLs of the files in the provisioning file must be resolved using this DNS server or the DNS server listed in the DHCP offer.

Up to three DNS addresses (primary, secondary, tertiary) can be listed in the DNS sections of the provisioning file. If all three DNS addresses are listed, they are attempted in the listing order to resolve the file URLs.

7.4.5.3 File download and redundancy

Up to three locations can be set for each file type, using the `primary-url`, `secondary-url`, and `tertiary-url` fields. The auto-provisioning process attempts to download all files using the `primary-url` information for each file. If this attempt is unsuccessful, the process reattempts using the `secondary-url` information for each file. If this attempt is not successful, the process reattempts using the `tertiary-url` information.

A ZTP operation is considered successful when all files named in the provisioning file are downloaded. If all file locations are attempted and all named files are not successfully downloaded, the auto-provisioning process fails and ZTP reattempts the provisioning process using the next valid interface.

7.4.5.4 Configuring the ZTP timeout in the provisioning file

The ZTP timeout is the total time allowed for the ZTP process to execute successfully. If the ZTP process fails to run successfully within the ZTP timeout duration, the node is rebooted and the ZTP process is retried. The default timeout is 30 minutes.

See [Options and option modification](#) for information about how to configure the ZTP timeout using CLI commands.

ZTP timeout information can be included in the provisioning file to configure a value different from the default. For a provisioning file example, see [Example provisioning file](#).

Example

The following example shows how the timeout is added to the provisioning file to change the default value.

```
set {
  timeout hours 1
  timeout minutes 30
}
```

7.4.5.5 Downloading the image file

The image file can be downloaded in the following ways:

- Extract the images from the OLCS CFLASH file and explicitly list them in the provisioning file. Each of the `.tim` files is validated and the version of the software is checked. The hash-value must be explicitly listed.

```
download {
  image "${B00T-PATH}/both.tim" {
```

```

    primary-url "${FILESERVER}/both.tim"
    verification {
        hash-algorithm sha256
        hash-value "53473e2727caf55f3a38fa466622af2147762e26a8587e9248240a572cdee849"
    }
}
.....
}

```

- Zip the image files and perform a single image download. ZTP can extract the images and verify the .tim file and software version. An additional file can also be listed for all files to be verified using sha256 or md5.

```

download {
    image "${B00T-PATH}/images.zip" {
        primary-url "${FILESERVER}/images.zip"
        unzip
        verification {
            hash-algorithm sha256
            hash-value "53473e2727caf55f3a38fa466622af2147762e26a8587e9248240a572cdee849"
        }
    }
    # SHA-256 Checksum Files - File of a list of SHAR256 checksum values and file names (must
    be absolute)
    sha256 "${B00T-PATH}/somefile.txt" {
        primary-url "${FILESERVER}/sha256-checksums.txt"
        verification {
            hash-algorithm sha256
            hash-value "53473e2727caf55f3a38fa466622af2147762e26a8587e9248240a572cdee849"
        }
    }
}
}

```

7.4.5.6 Example provisioning file

This section provides examples of provisioning file information.

Example: Provisioning file information

```

set {
    hours 1
    minutes 30
}
dns {
    primary 192.0.2.1
    secondary 192.0.2.2
    tertiary 192.0.2.3
    domain sample.domain.com
}
download {
    image "cf3:/both.tim" {
        primary-url "http://192.168.40.140:81/both.tim"
        secondary-url "http://192.168.40.140:81/both.tim"
        tertiary-url "http://192.168.40.140:81/both.tim"
    }
    image "cf3:/support.tim" {
        primary-url "http://192.168.40.140:81/support.tim"
        secondary-url "http://192.168.40.140:81/support.tim"
        tertiary-url "http://192.168.40.140:81/support.tim"
    }
}

```

```

}
config "cf3:/config.cfg" {
    primary-url "ftp://ftpserv:name@192.168.194.50/./images/dut-a.cfg"
    secondary-url "http://192.168.41.140:81/dut-a.cfg"
    tertiary-url "http://192.168.42.140:81/dut-a.cfg"
}
file "cf3:/license.txt" {
    primary-url "ftp://ftpserv:name@192.168.194.50/./images/provision_example.cfg"
    secondary-url "http://192.168.41.140:81/dut-a.cfg"
    tertiary-url "http://192.168.42.140:81/dut-a.cfg"
}
}
bof {
    primary-image cf3:/both.tim
    primary-config cf3:/config.tim
    address 192.168.100.1 active
    autonegotiate
    duplex full
    speed 100
    wait 3
    persist off
    console-speed 115200
}

```

For an HTTPS URL, the trust anchor needs to be referenced in the provisioning file. The trust anchor name references the entry in the `import trust-anchor` section of the file.

In the following example, the trust anchor name is `TRUST_ANCHOR`.

Example: Trust anchor for HTTPS URL in provisioning file

```

import {
    client {
        cert "cf3:/client.crt" {
            format pem
            primary-url http://10.10.10.67:81/client.crt
        }
        key "cf3:/client.key" {
            format pem
            primary-url http://10.10.10.67:81/client.key
        }
    }
    trust-anchor TRUST_ANCHOR{
        cert "cf3:/ca.crt" {
            format pem
            primary-url ftp://user-name:password@10.10.10.66//user-name/logs/
fileserver-4/ca.crt
        }
        crl "cf3:/ca.crl" {
            format der
            primary-url ftp://user-name:password@10.10.10.66//user-name/logs/
fileserver-4/ca.crl
        }
    }
}
<snip>
download {
    config "cf3:/ztp/ztp_dut-a.cfg" {
        primary-url "https://10.10.10.64:81/ztp_node-2.cfg"
        primary-trust-anchor "TRUST_ANCHOR"
    }
}

```

7.4.5.7 Proxy support

HTTP and HTTPS can connect to public servers using a proxy. The proxy is in URL format and the URL must be resolved using the provisioning file DNS.

The proxy can include a username and password. Proxy Auto-Configuration (PAC) is not supported.

Proxy information formatting is as follows:

`http://user@hostname:file-path`

`https://user@hostname:file-path`

proxy `http://ip-or-url user@hostname:port`

The HTTP (or HTTPS) proxy support information is included in **file** commands and in the ZTP provisioning file. The following example shows HTTP proxy information in the provisioning file.

Example

```
image "cf3:/both.tim" {
  primary-url "http://200.150.40.140:81/both.tim"
  secondary-url "http://200.150.40.140:81/both.tim"
  tertiary-url "http://200.150.40.140:81/both.tim"
  primary-proxy http://132.2.3.1:8080
  secondary-proxy http://133.3.4.1:8080
}
```

7.4.6 Day 0 configuration

To improve initial router functionality and reuse information discovered by ZTP in the configuration file, the user can include an optional day 0 configuration template within the provisioning file. This day 0 configuration template can obtain configuration details for specific modules that can be used to provision the node and establish node connectivity. Predefined symbols are used to add parameters discovered by ZTP, such as the port and VLAN, to the configuration template.

Any supported configuration can be included in the day 0 configuration template, including configuration of port types and services. However, Nokia recommends that day 0 configuration details should be kept to the minimum necessary to ensure that basic network connectivity is available for the ZTP process to run and complete.



Caution: Before using the configuration, verify the configuration details in the day 0 template and check for syntax, context, and other errors. The ZTP process does not verify the configuration details before implementing them.

When the provisioning file is received through the ZTP process, the day 0 configuration template section of the provisioning file is converted to a configuration file, as specified by the **Write** `cf3:/configuration-file-name.cfg` entry within the template. The BOF section of the provisioning file must specify the configuration file generated from the day 0 configuration template using the `primary-config cf3:/configuration-file-name.cfg` entry.

Some unknown elements within the day 0 configuration template can be entered as symbols, which are then converted to discovered information as the node is provisioned; see [Day 0 symbols](#).

See [Sample day 0 configuration template](#) for a configuration example, including the day 0 configuration template and associated BOF section of the provisioning file.

7.4.6.1 Day 0 configuration for multi-slot routers

On multi-slot routers, the discovered IOMs, MDAs, IMMs, and fabric (that is, the discovered hardware) are part of the day 0 configuration. ZTP automatically adds the CLI configuration to the correct location in the day 0 configuration in the provisioning file.

This process ensures that the nodes for different sites are populated with the correct hardware, regardless of what physical hardware is used for a specific site.

The day 0 configuration process for multi-slot routers is as follows:

1. The provider creates a day 0 configuration template with the required services and interfaces.
2. ZTP provisions the IOMs, MDAs, IMMs, and fabric.
3. ZTP discovers the port and VLAN on which the DHCP server is connected.
4. ZTP adds the discovered and provisioned hardware to the day 0 configuration in the provisioning file
5. ZTP uses the day 0 configuration in provisioning file to create an initial configuration file and saves it to the specified location on the node.

The BOF section of the provisioning file must specify the configuration file generated from the day 0 configuration.

6. After rebooting, the node boots up using the day 0 configuration file.

7.4.6.2 Day 0 symbols

Use symbols to generalize and reuse a single provisioning file across multiple ZTP deployments and for value substitutions and conditions in the day 0 configuration template of the provisioning files. The same symbol can appear as many times as required and can be used for both conditional statements and substitutions.

7.4.6.2.1 Symbol use for substitution

A day 0 configuration template can include symbols that are replaced with discovered information when the day 0 configuration is implemented. Symbols for substitution must be entered using the format `$(<symbol-name>)`. The following example shows a supported use case of a symbol for substitution.

Example

```
card $(ztp.bootstrap.slot)
  card-type $(ztp.bootstrap.card-type)
  mda $(ztp.bootstrap.mda)
  mda-type $(ztp.bootstrap.mda-type)
exit
exit
```

7.4.6.2.2 Symbol use for conditions

A day 0 configuration template can include symbols that are replaced with discovered information when the day 0 configuration is implemented. Symbols for conditions must be entered using the format `@(<symbol-name>)` at the beginning of a line. If the condition is not met (that is, the symbol does not exist), the rest

of the line is skipped and not written into the day 0 configuration file on the compact flash. The following example shows a supported use case of a symbol for conditions.

Example

```
@(ztp.uplink.connector)    port (ztp.uplink.connector)
@(ztp.uplink.connector)    connector
@(ztp.uplink.connector)    breakout $(ztp.uplink.breakout)
@(ztp.uplink.connector)    exit
@(ztp.uplink.connector)    exit
```

7.4.6.2.3 Supported symbols

The following table describes the supported symbols for a day 0 configuration template.

Table 19: Supported symbols

Symbol name	Description	Example value	Conditional	Discovered
sys.platform	Platform name	"7705"	No. Always available.	Configured for the system and is exposed at system boot-up
sys.type	Chassis type	"SAR-1"	No. Always available.	Configured for the system and is exposed at system bootup
sys.serial-number	Serial number of the chassis	"NS1234567"	No. Always available.	Configured for the system and is exposed at system bootup
sys.mac-address	Base chassis MAC of chassis	"01:01:01:01:01:01"	No. Always available.	Configured for the system and is exposed at system bootup
ztp.bootstrap.uplink	Full uplink used for bootstrap (including VLAN). Intended to be used as interface port binding in the day 0 configuration.	"1/1/1" "1/1/1:1000" "A/1"	No. Always available.	Discovered at VLAN discover or ZTP discovery time
ztp.bootstrap.uplink.card-type	Card-type of uplink slot used for bootstrap	"xcm-1s"	Yes. Available when uplink is inband.	Discovered at VLAN discover or ZTP discovery time

Symbol name	Description	Example value	Conditional	Discovered
ztp.bootstrap.uplink.slot	Slot of uplink used for bootstrap	"1"	Yes. Available when uplink is inband.	Discovered at VLAN discover or ZTP discovery time
ztp.uplink.xiom	Xiom identifier of uplink used for bootstrap	"x1"	Yes. Available when uplink is inband and has an xiom.	Discovered at VLAN discover or ZTP discovery time
ztp.uplink.xiom-type	Xiom-type of xiom used for bootstrap	lom2-se-6.0t"	Yes. Available when uplink is inband and has an xiom.	Discovered at VLAN discover or ZTP discovery time
ztp.uplink.mda	The MDA number of the MDA used for bootstrap (This will either be the MDA under card or MDA under xiom if uplink has xiom)	"1"	Yes. Available when uplink is inband.	Discovered at VLAN discover or ZTP discovery time
ztp.uplink.mda-type	The MDA number of the MDA used for bootstrap (This will either be the MDA under card or MDA under xiom if uplink has xiom)	"m4-sfp"	Yes. Available when uplink is inband.	Discovered at VLAN discover or ZTP discovery time
ztp.bootstrap.uplink.port	Uplink port without encapsulation value (VLAN)	1/2/c1/1	No. Always available.	Discovered at VLAN discover or ZTP discovery time
ztp.bootstrap.uplink.vlan	Uplink VLAN (encapsulation value)	"1000"	Yes. Available when uplink has a vlan.	Discovered at VLAN discover or ZTP discovery time
ztp.bootstrap.connector	Uplink connector ID	1/1/c1	Yes. Available when uplink is a connector-port.	Discovered at VLAN discover or ZTP discovery time
ztp.bootstrap.breakout	Connector breakout type	c1-10g	Yes. Available when uplink is a connector port.	Discovered at VLAN discover or ZTP discovery time
ztp.bootstrap.rs-fec	Connector RS-FEC mode setting		Yes. If connector has a non-default	Discovered at VLAN discover or ZTP discovery time

Symbol name	Description	Example value	Conditional	Discovered
			rs-fec mode setting.	
ztp.bootstrap.uplink.rs-fec	Uplink port RS-FEC mode setting		Yes. If uplink port has a non-default rs-fec mode setting.	Discovered at VLAN discover or ZTP discovery time
ztp.bootstrap.uplink.speed	Uplink port speed setting		Yes. If uplink port has a non-default speed setting	Discovered at VLAN discover or ZTP discovery time
Ztp.bootstrap.ip	IP address used during bootstrap (IPv4 or IPv6)	1.2.3.4 1::344:2	No. Always available.	Discovered via DHCP offer
Ztp.bootstrap.prefixlen	The prefix length of the IP address used under interface during bootstrap		No. Always available.	Discovered via DHCP offer

7.4.6.3 Sample day 0 configuration template

The following is a sample of a day 0 configuration template included in a provisioning file.

Example

```
# Generate Day-0 Configuration
# -----
Write "cf3:/config-template.cfg" {
#####
# !!DAY0 CONFIG START!! #
#####
exit all
configure
#-----
echo "System Configuration"
#-----
    system
        name "chassis-name"
        snmp
        exit
        login-control
            idle-timeout disable
        exit
        time
            ntp
                server 192.168.194.202
                no shutdown
            exit
        exit
        thresholds
```

```

        rmon
        exit
    exit
exit
#-----
echo "System Security Configuration"
#-----
    system
        security
            telnet-server
            ftp-server
            snmp
                community "private" rwa version both
                community "public" r version both
            exit
        exit
    exit
#-----
echo "Log Configuration"
#-----
    log
        snmp-trap-group 90
        trap-target ""
    exit
    log-id 90
        from main change
        to snmp
    exit
exit
exit all
configure
router
#-----
echo "IP Configuration"
#-----
    interface "IF-1/1/c1/1"
        port ${ztp.bootstrap.uplink}
        address 192.168.0.1/31
        ipv6
        exit
        no shutdown
    exit
exit
exit all
#####
# !!DAY0 CONFIG END!! #
#####
}
# Generate New BOF File
# -----
bof {
    primary-image cf3:/image.both
    primary-config cf3:/config-template.cfg
    autonegotiate
    duplex full
    speed 100
    wait 3
    persist off
    console-speed 115200
}

```

7.4.7 Logs and events

ZTP displays detailed events about all stages of the auto-boot and auto-provisioning processes. All events are saved in a log file on the node at the end of the ZTP process.

7.5 SZTP

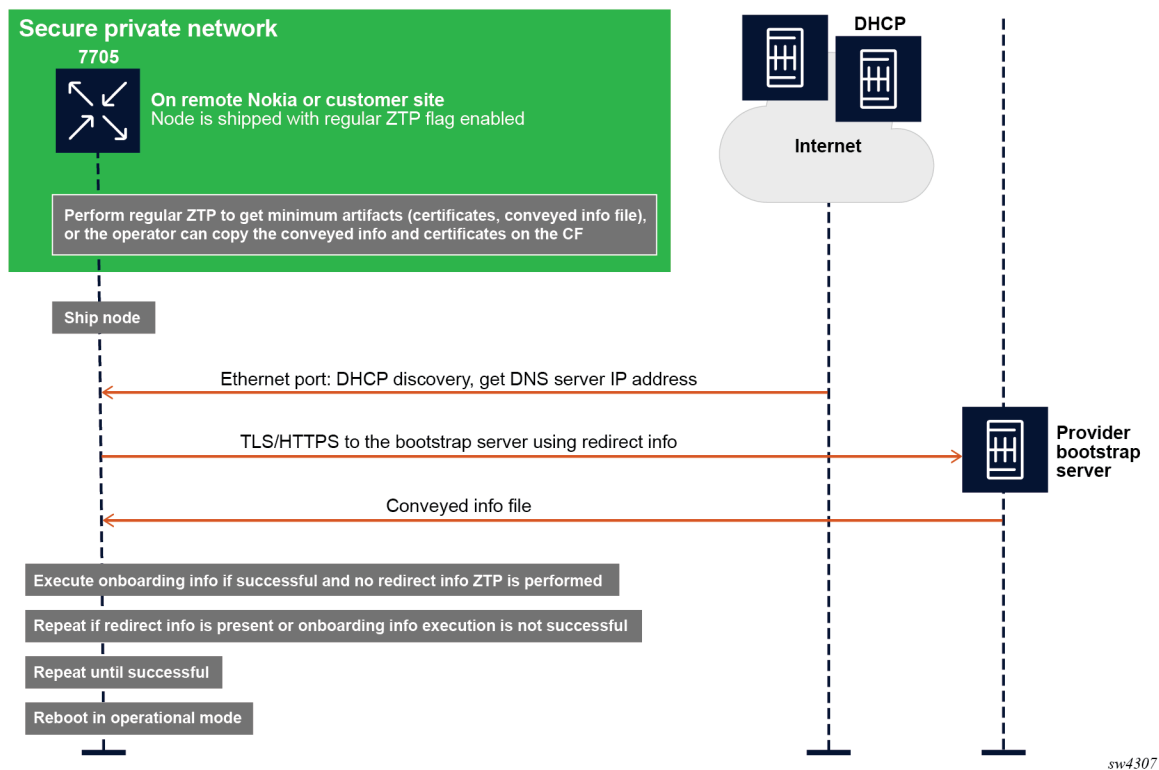
The SR OS implementation of SZTP is a partial application of RFC 8572 and is evolving to meet all RFC 8572 aspects. SZTP is an extension of ZTP as follows:

- The provisioning file and the node discovery of the MDAs, IOMs, and ports with links up are supported in both ZTP and SZTP.
- The ports that are ZTP-capable are also SZTP-capable.

SZTP securely bootstraps the node and provides it with the information required to boot up the node in an operational mode; this information includes all the initial artifacts required to create a mutual trust relationship between the node and the bootstrap server. After the node boots, it discovers the bootstrap server IP address, communicates with the server, and authenticates both the server and itself. Finally, the node securely downloads the encrypted boot image and initial configuration information.

SR OS uses different bootstrapping methods to obtain the required TLS certificates, trust anchors, and redirect information to connect securely to the server and download all the necessary information to boot in an operational mode.

Figure 13: SZTP process



In the example shown in the preceding figure, one of the following methods can be used to bootstrap the node securely:

- The operator stages the node at their own site and bootstraps it using ZTP through a secure or private network. The node obtains the TLS certificates, trust anchors, and keys from a conveyed information file, which must be copied into the compact flash (CF). The Uniform Resource Identifier (URI) of this file is included in the ZTP provisioning file.
- If the node has CF, the operator copies manually to the CF the certificates, trust anchors, and conveyed information file. Optionally, the operator can also include redirect information in the conveyed information file.

After the node is bootstrapped securely, it is shipped to the installation site, where it boots.

If the node has redirect information, it tries to connect the bootstrap server specified in the redirect information and establish a TLS session to create mutual trust between the node and the server.

If the node does not have redirect information, it performs a DHCP discovery and tries to obtain the redirect information using DHCP option 143 (IPv4) or 136 (IPv6). After obtaining the redirect information, the node tries to connect to the bootstrap server using TLS.

The node uses option 67 from the DHCP server or the URI from the file field of the redirect information to locate the conveyed information from the bootstrap server. The conveyed information provides the node with one of the following:

- more redirect information for a new file server and other required resources to connect to the file server to download all the required information and files

- onboarding information containing the URI of the boot image and the initial configuration
- both redirect information and onboarding information, in which case the node executes the onboarding information first and then executes the redirect information

7.5.1 Staging the secure environment

The following artifacts are required:

- node client certificates and keys
- bootstrap server certificates for the first redirect
- security artifacts file, which contains the trust anchor definitions and import instructions. Optionally, this file can also contain conveyed information, which consists of redirect, if applicable, and onboarding information.

The staging options are the following:

- Copy the artifacts to a CF that can be installed in `cf1:`, `cf2:`, or `cf3:` when the node is at the installation site.
- Alternatively, use ZTP to pre-stage the node and download the root security artifacts using a trusted DHCP server and provisioning file.



Note: Nokia recommends that the TLS client should have a certificate so that it is authenticated by the ZTP server. Although the client certification can be omitted for TLS configuration, this is not recommended.

7.5.2 Bootstrapping methods

The following bootstrapping methods are supported:

- Use DHCP option 143 (IPv4) or 163 (IPv6), as described in RFC 8572. Optionally, the operator can obtain the specific node URI (server and directory) by providing the DHCP server option 61 for the DHCP server, which in turn provides option 67 for the file directory, or option 143 (IPv4) or 163 (IPv6) with the server IP and file directory information. In this case, the TLS certificates, trust anchors, and keys must be installed on the node at the operator premises.
- Copy the following information to the CF:
 - redirect information for the bootstrap server
 - TLS certificates and trust anchors, and private keys
 - onboarding information
- Use ZTP to provide the following information to the node:
 - redirect information for the bootstrap server
 - TLS certificates and trust anchors, and private keys
 - onboarding information
- Redirect the node from the first bootstrap server to consecutive bootstrap servers. The bootstrap server can provide the node with additional redirect information in a secure encrypted manner. The redirect and onboarding information are provided in the conveyed information file.
 - redirect information to another bootstrap server

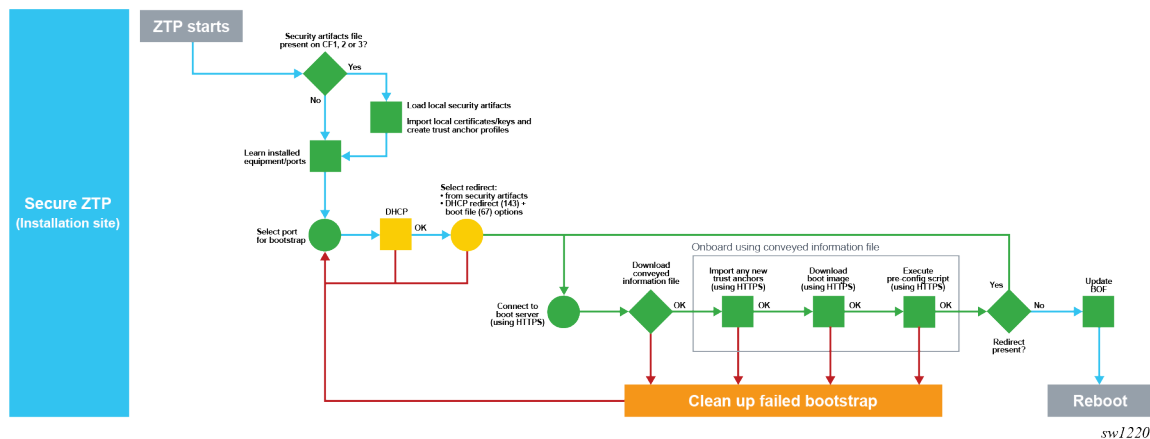
- required TLS certificates and trust anchors, and private keys
- onboarding information

7.5.3 Installation site process

At the installation site, the **auto-boot** flag in the BOF signals the ZTP process. The presence of a conveyed information file on the node signals to the node that it is a secure ZTP procedure.

The following figure shows the SZTP process at the installation site.

Figure 14: Installation site SZTP process



After staging, the port that has the link up is selected and SZTP is executed on it.

The node loads the security artifacts to install the TLS certificates and trust anchors. DHCP discovery messages are sent out on each port in sequence. If no DHCP offer is received, SZTP moves to the next port with the link up. The OOB port is examined first, followed by the untagged in-band ports. If no DHCP offer is received, VLAN discovery is performed on the in-band ports only by flooding VLAN 0 to 4196 with DHCP discovery.

After DHCP discovery completes, the node obtains an IP address and can optionally obtain option 143 (IPv4) or option 136 (IPv6) for redirect information. If the redirect information is present in the conveyed information file, it is preferred over the DHCP redirect information.

The node is connected to the bootstrap server as indicated by the redirect information and a TLS mutual authentication is established using the certificates. The bootstrap server must have the correct certificates, keys, and trust anchors to create the mutual TLS trust.

After the node authenticates the server and authenticates itself to the server, it downloads the conveyed information file from the server using HTTPS. The node obtains the server location of the conveyed information from DHCP option 67 or the file field in the redirect information.

If the conveyed information file contains redirect information, the node tries to connect to the new bootstrap server indicated in the new redirect information. The node can download the new certificates indicated in the conveyed information.

If the conveyed information file contains only onboarding information, the node downloads the onboarding file.

If the conveyed information file contains both onboarding and redirect information to the next bootstrap server, the node executes the onboarding information first, then the redirect information.

The process is successful if the node executes the onboarding information without errors.

7.5.3.1 Initial conveyed information file

The conveyed information file (also referred to as `conveyed-info.ztp` file) contains the certificates, keys, and trust anchors required to establish the TLS connection. This is the minimum information that the node requires to start SZTP after staging at the installation site. The initial file must be added to `cf3`: by copying it on the CF manually or using regular ZTP procedures and the provisioning file.

Example: Contents of a conveyed information file

```
import {
  client {
    cert "cf3:/artifacts/node.cert"
    key "cf3:/artifacts/node.key" {
      format der
    }
  }
  trust-anchor BOOTSERVER {
    cert "cf3:/artifacts/bootserver.cert"
  }
}
```

The certificates, keys, and trust anchor information can be encrypted using the **encrypt** command, as shown in the following example. When the **encrypt** keyword is present, the information is downloaded from the URI and encrypted using AES256.

Example: Using the encrypt command

```
import {
  client {
    encrypt
    cert "cf3:/artifacts/node.cert"
    key "cf3:/artifacts/node.key" {
      format der
    }
  }
  trust-anchor BOOTSERVER {
    encrypt
    cert "cf3:/artifacts/bootserver.cert"
  }
}
```

Optionally, the file can contain the redirect information as shown in the following example. It is not mandatory to include the redirect information in the file because the preliminary redirect information can be obtained using DHCP.



Note: The redirect information in the file is preferred over the DHCP redirect information because it is trusted.

Example: Redirect information in the file

```
import {
  client {
```

```

    encrypt
    cert "cf3:/artifacts/node.cert"
    key "cf3:/artifacts/node.key" {
        format der
    }
}
trust-anchor BOOTSERVER {
    encrypt
    cert "cf3:/artifacts/bootserver.cert"
}
}

redirect-information {
    boot-server "https://mybootserver.com/" {
        port 50
        trust-anchor BOOTSERVER
        file "conveyed.info"
    }
    boot-server "https://backupserver.com/" {
        port 50
        trust-anchor BOOTSERVER
        file "conveyed.info"
    }
}
}

```

The following example shows a file containing the entire conveyed information, including redirect and onboarding information. See [Onboarding information](#).

Example: File with entire conveyed information

```

import {
    client {
        encrypt
        cert "cf3:/artifacts/node.cert"
        key "cf3:/artifacts/node.key" {
            format der
        }
    }
    trust-anchor BOOTSERVER {
        encrypt
        cert "cf3:/artifacts/bootserver.cert"
    }
}

redirect-information {
    boot-server "https://mybootserver.com/" {
        port 50
        trust-anchor BOOTSERVER
        file "conveyed.info"
    }
    boot-server "https://backupserver.com/" {
        port 50
        trust-anchor BOOTSERVER
        file "conveyed.info"
    }
}

onboarding-information {
    boot-image
    download-uri https://images.com/$(sys.platform).zip
    pre-configuration-script "https://config.com/provisioning.cfg"
}

```

7.5.3.2 Onboarding information

The onboarding information is required to obtain all critical resources to boot the node in the normal mode of operation with the latest boot image and configuration. When processing the onboarding information, the device must first process the boot image information (if any), then execute the preconfiguration script (if any).

The onboarding information is present in the conveyed information file only. See [Conveyed information](#).

Example: Onboarding information

```
onboarding-information {
  boot-image
    download-uri https://images.com/$(sys.platform).zip
  pre-configuration-script "https://config.com/provisioning.cfg"
}
```

The download-uri is the URI to the boot image in ZIP format only. A URI list can exist, each pointing to the primary, secondary, or tertiary images.zip file. SR OS has multiple images, for example, CPM image and IOM image, and these images can be downloaded in a ZIP file. The URI can point to the ZIP file bundle to download all the images from a primary source. If the image is downloaded using download-uri, the destination of the image is always cf3:.

Example: Using download-uri information

```
onboarding-information {
  boot-image {
    # Download-URI(Mandatory): URL to ZIP file of images. Up to 3 for primary,
    secondary, tertiary
    # -> Base directory is "cf3:/"
    download-uri "https://server.download.com/images.zip"
    download-uri "https://server2.download.com/images.zip"
    download-uri "https://server3.download.com/images.zip"
    # VERIFICATION (Optional): File within ZIP file to verify images
    image-verification {
      hash-algorithm sha256
      hash-value "53473e2727caf55f3a38fa466622af2147762e26a8587e9248240a572cdee849"
    }
  }

  # Another provisioning file that can be loaded. Downloaded as "cf3:/autoboot.cfg.1,2,3"
  based on chaining
  pre-configuration-script "https://server.file.com/someotherprovisioningfile.txt"
}
```

Optionally, the node can run a checksum on the downloaded ZIP file using a hash to ensure there are no download errors. The hash algorithm and the hash value are noted in the onboarding information, as shown in the previous onboarding information example. The supported hash algorithms are SHA256 and MD5.

The preconfiguration script can be used to download the provisioning file. The provisioning file and the ZTP provisioning file have the same format. The provisioning file has to be executed to completion for the ZTP process to be successful. The provisioning file can also contain the location of the image. The image can be downloaded using the download URI or the provisioning file. When downloading the image using the provisioning file, the destination of the image can be dictated. The BOF must be configured accordingly to boot from the image destination. The image destination must always be cf3: or a folder in cf3:.



Note: The preconfiguration script is always required to clear the **auto-boot** flag from the BOF. A minimal BOF configuration is required in the provisioning file.

7.5.3.2.1 Preconfiguration script

The preconfiguration script is the actual ZTP provisioning file. SZTP supports all features of the ZTP provisioning file.

The preconfiguration script and provisioning file must be executed by the onboarding information to update the BOF configuration and remove the **auto-boot** flag. This ensures that the node comes back up in a normal mode of operation. For examples of the preconfiguration script and provisioning file information included in the onboarding information, see [Onboarding information](#).

7.5.3.2.2 Additional capabilities in the ZTP provisioning file

The following optional capabilities of the provisioning file are supported for both ZTP and SZTP:

- **checksum for file download**

SHA256 and MD5 are supported. The hash algorithm and hash value can be specifically configured for the file as shown in the following example. The file checksum is checked against the hash algorithm and hash value.

```
# List of Configuration Files to download
config "$(BOOT-PATH)/somefile.txt" {
    primary-url "$(FILESERVER)/somefile.txt"
    verification {
        hash-algorithm sha256
        hash-value
        "53473e2727caf55f3a38fa466622af2147762e26a8587e9248240a572cdee849"
    }
}
```

- **file encryption using AES256**

When a file is downloaded and if the file has the **encrypt** keyword enabled in the provisioning file, the file is encrypted using AES256 and placed on the CF. This is useful when downloading certificates, keys, and trust anchors for TLS. The following example shows an excerpt from the provisioning file. In this example, in the certificate import the keyword **encrypt** is used to encrypt each file on the CF after the file was downloaded. In addition, the checksum is calculated on each file and checked to ensure the files are downloaded without errors.

```
import {
    client {
        cert "$(CERT-PATH)/device.crt" {
            primary-url "$(FILESERVER)/device.crt"
            encrypt
            verification {
                hash-algorithm sha256
                hash-value
                "53473e2727caf55f3a38fa466622af2147762e26a8587e9248240a572cdee849"
            }
        }
        key "$(CERT-PATH)/device.key" {
            format pem
            primary-url "$(FILESERVER)/device.key"
            encrypt
            verification {
```

```

        hash-algorithm sha256
        hash-value
        "53473e2727caf55f3a38fa466622af2147762e26a8587e9248240a572cdee849"
    }
}
trust-anchor "NokiaSvcS" {
    cert "$(CERT-PATH)/owner-ca.crt" {
        format pem
        primary-url "$(FILESERVER)/owner-ca.crt"
        encrypt
        verification {
            hash-algorithm sha256
            hash-value
            "53473e2727caf55f3a38fa466622af2147762e26a8587e9248240a572cdee849"
        }
    }
    crl "$(CERT-PATH)/owner-ca.crl"
    format der
    primary-url "$(FILESERVER)/owner-ca.crl"
    encrypt
    verification {
        hash-algorithm sha256
        hash-value
        "53473e2727caf55f3a38fa466622af2147762e26a8587e9248240a572cdee849"
    }
}
}
}

```

- **downloading the image files in ZIP file format**

The files can be downloaded using the provisioning file with an optional checksum. The ZIP file is automatically unzipped and each individual file is placed on the CF. The checksum is checked across the entire ZIP file before it is unzipped.



Note: The specified directory path must end with a forward slash (/).

```

image "$(BOOT-PATH)/images.zip" {
    primary-url "$(FILESERVER)/images.zip"
    unzip {
        directory "cf3:/ztp/"
    }
    verification {
        hash-algorithm sha256
        hash-value
        "53473e2727caf55f3a38fa466622af2147762e26a8587e9248240a572cdee849"
    }
}

```

- **downloading a SHA256 or MD5 checksum file**

The user can download a SHA256 or MD5 checksum file from Nokia servers for the ZIP image files. After the image file is unzipped, each *.tim file has its checksum checked against these checksum files.

The following are examples of checksum files.

```

17717f13ecf19179e367d990277e943993d53d771b44d19fddf5d349b1e7e7c4 cf3:/ztp/cpm.tim
616656b2224e65e29d71355ea41d9dc548c281e9f729c97fe46f3a5c643acb09 cf3:/ztp/iom.tim
dd9455158dc2dfbf937eb372bbef73ebab97f951efa742eec16a4ed4edd3ca9b cf3:/ztp/support.tim

```

Checksum or file decryption errors cause the failure of the ZTP or SZTP procedure, in which case the node generates an error and logs the event.

7.5.3.2.3 Certificate chaining

The following is an example of certificate chaining using a root and intermediate CA in the provisioning file.

Example: Certificate chaining

```
import {
  trust-anchor CMP_ROOT {
    cert "cf3:/root.crt" {
      format pem
      primary-url "http://ztp-server.com/pki/certificates/Classified_Plattform_Root_
CA-1.pem"
      secondary-url "http://ztp-server.com/pki/certificates/Classified_Plattform_
Root_CA-1.pem"
    }
  }
  trust-anchor CMP_ISSUING {
    cert "cf3:/issuing.crt" {
      format pem
      primary-url "http://ztp-server.com/pki/certificates/Classified_Plattform_
Issuing_CA_1.pem"
      secondary-url "http://ztp-server.com/pki/certificates/Classified_Plattform_
Issuing_CA_1.pem"
    }
    crl "cf3:/issuing.crl" {
      format der
      primary-url "http://ztp-server.com/pki/crls/Classified_Plattform_Issuing_CA_
1.crl"
      secondary-url "http://ztp-server.com/pki/crls/Classified_Plattform_Issuing_CA_
1.crl"
    }
  }
}

download {
  config "cf3:/config.cfg" {
    primary-url "https://ztp-server.com/config-ztp-ne21.cfg"
    secondary-url "https://ztp-server.com/config-ztp-ne21.cfg"
    primary-trust-anchor "CMP_ISSUING"
    secondary-trust-anchor "CMP_ISSUING"
  }
  config "cf3:/snmp.cfg" {
    primary-url "https://ztp-server.com/snmp.cfg"
    secondary-url "https://ztp-server.com/snmp.cfg"
    primary-trust-anchor "CMP_ISSUING"
    secondary-trust-anchor "CMP_ISSUING"
  }
}

bof {
  primary-image cf3:\TiMOS-SR-24.7.R1
  primary-config cf3:\config.cfg
  wait 3
  persist on
}
```

7.5.3.3 Conveyed information

After SR OS authenticates successfully to the bootstrap server, the node can download the conveyed information using HTTPS. The operator can choose the name of the conveyed information file.

The SR OS conveyed information is trusted and does not require an additional signature verification.

Example: File with only onboarding information

The following conveyed information file example contains only onboarding information.

```
onboarding-information {
  boot-image
    download-uri https://images.com/$(sys.platform).zip
  pre-configuration-script "https://config.com/provisioning.cfg"
}
```

The conveyed information can also contain redirect information, in which case a recursive redirect can happen to another bootstrap server. If the conveyed information contains onboarding information and redirect information, the node executes the onboarding information first, then the redirect to the next bootstrap server.

Example: File with onboarding and redirect information

The following conveyed information file example contains onboarding and redirect information, and the certificates required for the second redirect.

```
onboarding-information {
  boot-image
    download-uri https://images.com/$(sys.platform).zip
  pre-configuration-script "https://config.com/provisioning.cfg"
}
import {
  client {
    cert "cf3:/artifacts/node.cert"
    key "cf3:/artifacts/node.key" {
      format der
    }
  }
  trust-anchor BOOTSERVER {
    cert "cf3:/artifacts/bootserver.cert"
  }
}

redirect-information {
  boot-server "https://mybootserver.com/" {
    port 50
    trust-anchor BOOTSERVER
    file "conveyed.info"
  }
  boot-server "https://backupserver.com/" {
    port 50
    trust-anchor BOOTSERVER
    file "conveyed.info"
  }
}
```

After the conveyed information is executed successfully, the BOF is loaded in the provisioning file to which the preconfiguration script is pointing and the **auto-boot** flag is cleared.

8 Standards and protocol support

**Note:**

The information provided in this chapter is subject to change without notice and may not apply to all platforms.

Nokia assumes no responsibility for inaccuracies.

8.1 Bidirectional Forwarding Detection (BFD)

RFC 5880, *Bidirectional Forwarding Detection (BFD)*

RFC 5881, *Bidirectional Forwarding Detection (BFD) IPv4 and IPv6 (Single Hop)*

RFC 5882, *Generic Application of Bidirectional Forwarding Detection (BFD)*

8.2 Border Gateway Protocol (BGP)

draft-hares-idr-update-attr-low-bits-fix-01, *Update Attribute Flag Low Bits Clarification*

draft-ietf-idr-add-paths-guidelines-08, *Best Practices for Advertisement of Multiple Paths in IBGP*

draft-ietf-idr-best-external-03, *Advertisement of the best external route in BGP*

draft-ietf-idr-bgp-gr-notification-01, *Notification Message support for BGP Graceful Restart*

draft-ietf-idr-bgp-ls-app-specific-attr-16, *Application-Specific Attributes Advertisement with BGP Link-State*

draft-ietf-idr-bgp-optimal-route-reflection-10, *BGP Optimal Route Reflection (BGP-ORR)*

draft-ietf-idr-error-handling-03, *Revised Error Handling for BGP UPDATE Messages*

draft-ietf-idr-link-bandwidth-03, *BGP Link Bandwidth Extended Community*

RFC 1772, *Application of the Border Gateway Protocol in the Internet*

RFC 1997, *BGP Communities Attribute*

RFC 2385, *Protection of BGP Sessions via the TCP MD5 Signature Option*

RFC 2439, *BGP Route Flap Damping*

RFC 2545, *Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing*

RFC 2858, *Multiprotocol Extensions for BGP-4*

RFC 2918, *Route Refresh Capability for BGP-4*

RFC 4271, *A Border Gateway Protocol 4 (BGP-4)*

RFC 4360, *BGP Extended Communities Attribute*

RFC 4364, *BGP/MPLS IP Virtual Private Networks (VPNs)*

RFC 4456, *BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)*

RFC 4486, *Subcodes for BGP Cease Notification Message*

RFC 4659, *BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN*

RFC 4684, *Constrained Route Distribution for Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs)*

RFC 4724, *Graceful Restart Mechanism for BGP – helper mode*

RFC 4760, *Multiprotocol Extensions for BGP-4*

RFC 4798, *Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE)*

RFC 5004, *Avoid BGP Best Path Transitions from One External to Another*

RFC 5065, *Autonomous System Confederations for BGP*

RFC 5291, *Outbound Route Filtering Capability for BGP-4*

RFC 5396, *Textual Representation of Autonomous System (AS) Numbers – asplain*

RFC 5492, *Capabilities Advertisement with BGP-4*

RFC 5668, *4-Octet AS Specific BGP Extended Community*

RFC 6286, *Autonomous-System-Wide Unique BGP Identifier for BGP-4*

RFC 6368, *Internal BGP as the Provider/Customer Edge Protocol for BGP/MPLS IP Virtual Private Networks (VPNs)*

RFC 6793, *BGP Support for Four-Octet Autonomous System (AS) Number Space*

RFC 6810, *The Resource Public Key Infrastructure (RPKI) to Router Protocol*

RFC 6811, *Prefix Origin Validation*

RFC 6996, *Autonomous System (AS) Reservation for Private Use*

RFC 7311, *The Accumulated IGP Metric Attribute for BGP*

RFC 7606, *Revised Error Handling for BGP UPDATE Messages*

RFC 7607, *Codification of AS 0 Processing*

RFC 7752, *North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP*

RFC 7911, *Advertisement of Multiple Paths in BGP*

RFC 7999, *BLACKHOLE Community*

RFC 8092, *BGP Large Communities Attribute*

RFC 8097, *BGP Prefix Origin Validation State Extended Community*

RFC 8212, *Default External BGP (EBGP) Route Propagation Behavior without Policies*

RFC 8277, *Using BGP to Bind MPLS Labels to Address Prefixes*

RFC 9494, *Long-Lived Graceful Restart for BGP*

8.3 Bridging and management

IEEE 802.1AB, *Station and Media Access Control Connectivity Discovery*

IEEE 802.1ad, *Provider Bridges*

IEEE 802.1AX, *Link Aggregation*

IEEE 802.1Q, *Virtual LANs*

8.4 Certificate management

RFC 4210, *Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)*
RFC 4211, *Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)*
RFC 5280, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*
RFC 6712, *Internet X.509 Public Key Infrastructure -- HTTP Transfer for the Certificate Management Protocol (CMP)*
RFC 7030, *Enrollment over Secure Transport*
RFC 7468, *Textual Encodings of PKIX, PKCS, and CMS Structures*

8.5 Ethernet VPN (EVPN)

draft-ietf-bess-bgp-srv6-args-00, *SRv6 Argument Signaling for BGP Services*
draft-ietf-bess-evpn-ipvpn-interworking-06, *EVPN Interworking with IPVPN*
draft-sr-bess-evpn-vpws-gateway-03, *Ethernet VPN Virtual Private Wire Services Gateway Solution*
RFC 7432, *BGP MPLS-Based Ethernet VPN*
RFC 8214, *Virtual Private Wire Service Support in Ethernet VPN*
RFC 8317, *Ethernet-Tree (E-Tree) Support in Ethernet VPN (EVPN) an Provider Backbone Bridging EVPN (PBB-EVPN)*
RFC 8365, *A Network Virtualization Overlay Solution Using Ethernet VPN (EVPN)*
RFC 8560, *Seamless Integration of Ethernet VPN (EVPN) with Virtual Private LAN Service (VPLS) and Their Provider Backbone Bridge (PBB) Equivalents*
RFC 9047, *Propagation of ARP/ND Flags in an Ethernet Virtual Private Network (EVPN)*
RFC 9135, *Integrated Routing and Bridging in Ethernet VPN (EVPN)*
RFC 9136, *IP Prefix Advertisement in Ethernet VPN (EVPN)*
RFC 9161, *Operational Aspects of Proxy ARP/ND in Ethernet Virtual Private Networks*
RFC 9251, *Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Proxies for Ethernet VPN (EVPN)*

8.6 gRPC Remote Procedure Calls (gRPC)

cert.proto version 0.1.0, *gRPC Network Operations Interface (gNOI) Certificate Management Service*
file.proto version 0.1.0, *gRPC Network Operations Interface (gNOI) File Service*
gnmi.proto version 0.8.0, *gRPC Network Management Interface (gNMI) Service Specification*
gnmi_ext.proto version 0.1.0, *gNMI Commit Confirmed Extension*
system.proto version 1.0.0, *gRPC Network Operations Interface (gNOI) System Service*
PROTOCOL-HTTP2, *gRPC over HTTP2*

8.7 Intermediate System to Intermediate System (IS-IS)

draft-ietf-isis-mi-02, *IS-IS Multi-Instance*

draft-ietf-lsr-igp-ureach-prefix-announce-01, *IGP Unreachable Prefix Announcement* – without U-Flag and UP-Flag

draft-kaplan-isis-ext-eth-02, *Extended Ethernet Frame Size Support*

ISO/IEC 10589:2002 Second Edition, *Intermediate system to Intermediate system intra-domain routing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473)*

RFC 1195, *Use of OSI IS-IS for Routing in TCP/IP and Dual Environments*

RFC 2973, *IS-IS Mesh Groups*

RFC 3359, *Reserved Type, Length and Value (TLV) Codepoints in Intermediate System to Intermediate System*

RFC 3719, *Recommendations for Interoperable Networks using Intermediate System to Intermediate System (IS-IS)*

RFC 3787, *Recommendations for Interoperable IP Networks using Intermediate System to Intermediate System (IS-IS)*

RFC 5120, *M-ISIS: Multi Topology (MT) Routing in IS-IS*

RFC 5130, *A Policy Control Mechanism in IS-IS Using Administrative Tags*

RFC 5301, *Dynamic Hostname Exchange Mechanism for IS-IS*

RFC 5302, *Domain-wide Prefix Distribution with Two-Level IS-IS*

RFC 5303, *Three-Way Handshake for IS-IS Point-to-Point Adjacencies*

RFC 5304, *IS-IS Cryptographic Authentication*

RFC 5305, *IS-IS Extensions for Traffic Engineering TE*

RFC 5306, *Restart Signaling for IS-IS – helper mode*

RFC 5308, *Routing IPv6 with IS-IS*

RFC 5309, *Point-to-Point Operation over LAN in Link State Routing Protocols*

RFC 5310, *IS-IS Generic Cryptographic Authentication*

RFC 6213, *IS-IS BFD-Enabled TLV*

RFC 6232, *Purge Originator Identification TLV for IS-IS*

RFC 6233, *IS-IS Registry Extension for Purges*

RFC 7775, *IS-IS Route Preference for Extended IP and IPv6 Reachability*

RFC 7794, *IS-IS Prefix Attributes for Extended IPv4 and IPv6 Reachability* – sections 2.1 and 2.3

RFC 7981, *IS-IS Extensions for Advertising Router Information*

RFC 7987, *IS-IS Minimum Remaining Lifetime*

RFC 8202, *IS-IS Multi-Instance* – single topology

RFC 8570, *IS-IS Traffic Engineering (TE) Metric Extensions* – Min/Max Unidirectional Link Delay metric for flex-algo, RSVP, SR-TE

RFC 8919, *IS-IS Application-Specific Link Attributes*

8.8 Internet Protocol (IP) general

draft-grant-tacacs-02, *The TACACS+ Protocol*

RFC 768, *User Datagram Protocol*

RFC 793, *Transmission Control Protocol*

RFC 854, *Telnet Protocol Specifications*

RFC 1350, *The TFTP Protocol (revision 2)*

RFC 2784, *Generic Routing Encapsulation (GRE)*

RFC 3164, *The BSD syslog Protocol*

RFC 4632, *Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan*

RFC 5082, *The Generalized TTL Security Mechanism (GTSM)*

8.9 Internet Protocol (IP) multicast

RFC 1112, *Host Extensions for IP Multicasting*

RFC 2236, *Internet Group Management Protocol, Version 2*

RFC 2365, *Administratively Scoped IP Multicast*

RFC 2375, *IPv6 Multicast Address Assignments*

RFC 2710, *Multicast Listener Discovery (MLD) for IPv6*

RFC 3376, *Internet Group Management Protocol, Version 3*

RFC 3446, *Anycast Rendezvous Point (RP) mechanism using Protocol Independent Multicast (PIM) and Multicast Source Discovery Protocol (MSDP)*

RFC 3590, *Source Address Selection for the Multicast Listener Discovery (MLD) Protocol*

RFC 3618, *Multicast Source Discovery Protocol (MSDP)*

RFC 3810, *Multicast Listener Discovery Version 2 (MLDv2) for IPv6*

RFC 4541, *Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches*

RFC 4604, *Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast*

RFC 4610, *Anycast-RP Using Protocol Independent Multicast (PIM)*

RFC 4611, *Multicast Source Discovery Protocol (MSDP) Deployment Scenarios*

RFC 5059, *Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM)*

RFC 5186, *Internet Group Management Protocol Version 3 (IGMPv3) / Multicast Listener Discovery Version 2 (MLDv2) and Multicast Routing Protocol Interaction*

RFC 5384, *The Protocol Independent Multicast (PIM) Join Attribute Format*

RFC 7761, *Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)*
RFC 8487, *Mtrace Version 2: Traceroute Facility for IP Multicast*

8.10 Internet Protocol (IP) version 4

RFC 791, *Internet Protocol*
RFC 792, *Internet Control Message Protocol*
RFC 826, *An Ethernet Address Resolution Protocol*
RFC 1034, *Domain Names - Concepts and Facilities*
RFC 1035, *Domain Names - Implementation and Specification*
RFC 1191, *Path MTU Discovery* – router specification
RFC 1519, *Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy*
RFC 1812, *Requirements for IPv4 Routers*
RFC 1918, *Address Allocation for Private Internets*
RFC 2131, *Dynamic Host Configuration Protocol*; Relay only
RFC 2132, *DHCP Options and BOOTP Vendor Extensions* – DHCP
RFC 2401, *Security Architecture for Internet Protocol*
RFC 3021, *Using 31-Bit Prefixes on IPv4 Point-to-Point Links*
RFC 3046, *DHCP Relay Agent Information Option (Option 82)*
RFC 3768, *Virtual Router Redundancy Protocol (VRRP)*
RFC 4884, *Extended ICMP to Support Multi-Part Messages* – ICMPv4 and ICMPv6 Time Exceeded

8.11 Internet Protocol (IP) version 6

RFC 2464, *Transmission of IPv6 Packets over Ethernet Networks*
RFC 2529, *Transmission of IPv6 over IPv4 Domains without Explicit Tunnels*
RFC 3122, *Extensions to IPv6 Neighbor Discovery for Inverse Discovery Specification*
RFC 3315, *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*
RFC 3587, *IPv6 Global Unicast Address Format*
RFC 3596, *DNS Extensions to Support IP version 6*
RFC 3633, *IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6*
RFC 3736, *Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6*
RFC 3971, *SEcure Neighbor Discovery (SEND)*
RFC 4007, *IPv6 Scoped Address Architecture*
RFC 4191, *Default Router Preferences and More-Specific Routes* – Default Router Preference
RFC 4193, *Unique Local IPv6 Unicast Addresses*

RFC 4291, *Internet Protocol Version 6 (IPv6) Addressing Architecture*
RFC 4443, *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*
RFC 4861, *Neighbor Discovery for IP version 6 (IPv6)*
RFC 5095, *Deprecation of Type 0 Routing Headers in IPv6*
RFC 5722, *Handling of Overlapping IPv6 Fragments*
RFC 5798, *Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6 – IPv6*
RFC 5952, *A Recommendation for IPv6 Address Text Representation*
RFC 6164, *Using 127-Bit IPv6 Prefixes on Inter-Router Links*
RFC 8021, *Generation of IPv6 Atomic Fragments Considered Harmful*
RFC 8200, *Internet Protocol, Version 6 (IPv6) Specification*

8.12 Internet Protocol Security (IPsec)

draft-ietf-ipsec-isakmp-mode-cfg-05, *The ISAKMP Configuration Method*
draft-ietf-ipsec-isakmp-xauth-06, *Extended Authentication within ISAKMP/Oakley (XAUTH)*
RFC 2401, *Security Architecture for the Internet Protocol*
RFC 2403, *The Use of HMAC-MD5-96 within ESP and AH*
RFC 2404, *The Use of HMAC-SHA-1-96 within ESP and AH*
RFC 2405, *The ESP DES-CBC Cipher Algorithm With Explicit IV*
RFC 2406, *IP Encapsulating Security Payload (ESP)*
RFC 2407, *IPsec Domain of Interpretation for ISAKMP (IPsec DoI)*
RFC 2408, *Internet Security Association and Key Management Protocol (ISAKMP)*
RFC 2409, *The Internet Key Exchange (IKE)*
RFC 2410, *The NULL Encryption Algorithm and Its Use With IPsec*
RFC 2560, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*
RFC 3526, *More Modular Exponential (MODP) Diffie-Hellman group for Internet Key Exchange (IKE)*
RFC 3566, *The AES-XCBC-MAC-96 Algorithm and Its Use With IPsec*
RFC 3602, *The AES-CBC Cipher Algorithm and Its Use with IPsec*
RFC 3706, *A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers*
RFC 3947, *Negotiation of NAT-Traversal in the IKE*
RFC 3948, *UDP Encapsulation of IPsec ESP Packets*
RFC 4106, *The Use of Galois/Counter Mode (GCM) in IPsec ESP*
RFC 4109, *Algorithms for Internet Key Exchange version 1 (IKEv1)*
RFC 4301, *Security Architecture for the Internet Protocol*
RFC 4303, *IP Encapsulating Security Payload*
RFC 4307, *Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)*

RFC 4308, *Cryptographic Suites for IPsec*
RFC 4434, *The AES-XCBC-PRF-128 Algorithm for the Internet Key Exchange Protocol (IKE)*
RFC 4543, *The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH*
RFC 4754, *IKE and IKEv2 Authentication Using the Elliptic Curve Digital Signature Algorithm (ECDSA)*
RFC 4835, *Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)*
RFC 4868, *Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec*
RFC 4945, *The Internet IP Security PKI Profile of IKEv1/ISAKMP, IKEv2 and PKIX*
RFC 5019, *The Lightweight Online Certificate Status Protocol (OCSP) Profile for High-Volume Environments*
RFC 5282, *Using Authenticated Encryption Algorithms with the Encrypted Payload of the IKEv2 Protocol*
RFC 5903, *ECP Groups for IKE and IKEv2*
RFC 5996, *Internet Key Exchange Protocol Version 2 (IKEv2)*
RFC 5998, *An Extension for EAP-Only Authentication in IKEv2*
RFC 6379, *Suite B Cryptographic Suites for IPsec*
RFC 6380, *Suite B Profile for Internet Protocol Security (IPsec)*
RFC 6960, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*
RFC 7296, *Internet Key Exchange Protocol Version 2 (IKEv2)*
RFC 7321, *Cryptographic Algorithm Implementation Requirements and Usage Guidance for Encapsulating Security Payload (ESP) and Authentication Header (AH)*
RFC 7383, *Internet Key Exchange Protocol Version 2 (IKEv2) Message Fragmentation*
RFC 7427, *Signature Authentication in the Internet Key Exchange Version 2 (IKEv2)*
RFC 8784, *Mixing Preshared Keys in the Internet Key Exchange Protocol Version 2 (IKEv2) for Post-quantum Security*

8.13 Label Distribution Protocol (LDP)

draft-pdutta-mpls-ldp-adj-capability-00, *LDP Adjacency Capabilities*
draft-pdutta-mpls-ldp-v2-00, *LDP Version 2*
RFC 3037, *LDP Applicability*
RFC 3478, *Graceful Restart Mechanism for Label Distribution Protocol – helper mode*
RFC 5036, *LDP Specification*
RFC 5283, *LDP Extension for Inter-Area Label Switched Paths (LSPs)*
RFC 5443, *LDP IGP Synchronization*
RFC 5561, *LDP Capabilities*
RFC 5919, *Signaling LDP Label Advertisement Completion*

8.14 Multiprotocol Label Switching (MPLS)

RFC 3031, *Multiprotocol Label Switching Architecture*

RFC 3032, *MPLS Label Stack Encoding*

RFC 3270, *Multi-Protocol Label Switching (MPLS) Support of Differentiated Services – E-LSP*

RFC 3443, *Time To Live (TTL) Processing in Multi-Protocol Label Switching (MPLS) Networks*

RFC 5332, *MPLS Multicast Encapsulations*

RFC 5884, *Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)*

RFC 6424, *Mechanism for Performing Label Switched Path Ping (LSP Ping) over MPLS Tunnels*

RFC 7308, *Extended Administrative Groups in MPLS Traffic Engineering (MPLS-TE)*

RFC 7746, *Label Switched Path (LSP) Self-Ping*

8.15 Network Address Translation (NAT)

RFC 4787, *Network Address Translation (NAT) Behavioral Requirements for Unicast UDP*

RFC 5382, *NAT Behavioral Requirements for TCP*

RFC 5508, *NAT Behavioral Requirements for ICMP*

8.16 Open Shortest Path First (OSPF)

RFC 1765, *OSPF Database Overflow*

RFC 2328, *OSPF Version 2*

RFC 3101, *The OSPF Not-So-Stubby Area (NSSA) Option*

RFC 3509, *Alternative Implementations of OSPF Area Border Routers*

RFC 3623, *Graceful OSPF Restart Graceful OSPF Restart – helper mode*

RFC 3630, *Traffic Engineering (TE) Extensions to OSPF Version 2*

RFC 4576, *Using a Link State Advertisement (LSA) Options Bit to Prevent Looping in BGP/MPLS IP Virtual Private Networks (VPNs)*

RFC 4577, *OSPF as the Provider/Customer Edge Protocol for BGP/MPLS IP Virtual Private Networks (VPNs)*

RFC 5185, *OSPF Multi-Area Adjacency*

RFC 5243, *OSPF Database Exchange Summary List Optimization*

RFC 5250, *The OSPF Opaque LSA Option*

RFC 5309, *Point-to-Point Operation over LAN in Link State Routing Protocols*

RFC 5642, *Dynamic Hostname Exchange Mechanism for OSPF*

RFC 6549, *OSPFv2 Multi-Instance Extensions*

RFC 6987, *OSPF Stub Router Advertisement*

RFC 7471, *OSPF Traffic Engineering (TE) Metric Extensions – Min/Max Unidirectional Link Delay metric for flex-algo, RSVP, SR-TE*

RFC 7684, *OSPFv2 Prefix/Link Attribute Advertisement*

RFC 7770, *Extensions to OSPF for Advertising Optional Router Capabilities*

RFC 8920, *OSPF Application-Specific Link Attributes*

8.17 Path Computation Element Protocol (PCEP)

draft-alvarez-pce-path-profiles-04, *PCE Path Profiles*

draft-ietf-pce-binding-label-sid-15, *Carrying Binding Label/Segment Identifier (SID) in PCE-based Networks*. – MPLS binding SIDs

RFC 5440, *Path Computation Element (PCE) Communication Protocol (PCEP)*

RFC 8231, *Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE*

RFC 8253, *PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)*

RFC 8281, *PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model*

RFC 8408, *Conveying Path Setup Type in PCE Communication Protocol (PCEP) Messages*

RFC 8664, *Path Computation Element Communication Protocol (PCEP) Extensions for Segment Routing*

8.18 Pseudowire (PW)

draft-ietf-l2vpn-vpws-iw-oam-04, *OAM Procedures for VPWS Interworking*

MFA Forum 12.0.0, *Multiservice Interworking - Ethernet over MPLS*

MFA Forum 13.0.0, *Fault Management for Multiservice Interworking v1.0*

MFA Forum 16.0.0, *Multiservice Interworking - IP over MPLS*

RFC 3916, *Requirements for Pseudo-Wire Emulation Edge-to-Edge (PWE3)*

RFC 3985, *Pseudo Wire Emulation Edge-to-Edge (PWE3)*

RFC 4385, *Pseudo Wire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN*

RFC 4446, *IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3)*

RFC 4447, *Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)*

RFC 4448, *Encapsulation Methods for Transport of Ethernet over MPLS Networks*

RFC 5085, *Pseudowire Virtual Circuit Connectivity Verification (VCCV): A Control Channel for Pseudowires*

RFC 5659, *An Architecture for Multi-Segment Pseudowire Emulation Edge-to-Edge*

RFC 5885, *Bidirectional Forwarding Detection (BFD) for the Pseudowire Virtual Circuit Connectivity Verification (VCCV)*

RFC 6073, *Segmented Pseudowire*

RFC 6310, *Pseudowire (PW) Operations, Administration, and Maintenance (OAM) Message Mapping*
RFC 6391, *Flow-Aware Transport of Pseudowires over an MPLS Packet Switched Network*
RFC 6575, *Address Resolution Protocol (ARP) Mediation for IP Interworking of Layer 2 VPNs*
RFC 6718, *Pseudowire Redundancy*
RFC 6829, *Label Switched Path (LSP) Ping for Pseudowire Forwarding Equivalence Classes (FECs) Advertised over IPv6*
RFC 6870, *Pseudowire Preferential Forwarding Status bit*
RFC 7023, *MPLS and Ethernet Operations, Administration, and Maintenance (OAM) Interworking*
RFC 7267, *Dynamic Placement of Multi-Segment Pseudowires*
RFC 7392, *Explicit Path Routing for Dynamic Multi-Segment Pseudowires – ER-TLV and ER-HOP IPv4 Prefix*
RFC 8395, *Extensions to BGP-Signaled Pseudowires to Support Flow-Aware Transport Labels*

8.19 Quality of Service (QoS)

RFC 2430, *A Provider Architecture for Differentiated Services and Traffic Engineering (PASTE)*
RFC 2474, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*
RFC 2597, *Assured Forwarding PHB Group*
RFC 3140, *Per Hop Behavior Identification Codes*
RFC 3246, *An Expedited Forwarding PHB (Per-Hop Behavior)*

8.20 Remote Authentication Dial In User Service (RADIUS)

draft-oscca-cfrg-sm3-02, *The SM3 Cryptographic Hash Function*
RFC 2865, *Remote Authentication Dial In User Service (RADIUS)*
RFC 2866, *RADIUS Accounting*
RFC 3162, *RADIUS and IPv6*
RFC 6613, *RADIUS over TCP – with TLS*
RFC 6614, *Transport Layer Security (TLS) Encryption for RADIUS*
RFC 6929, *Remote Authentication Dial-In User Service (RADIUS) Protocol Extensions*

8.21 Resource Reservation Protocol - Traffic Engineering (RSVP-TE)

RFC 2702, *Requirements for Traffic Engineering over MPLS*
RFC 2747, *RSVP Cryptographic Authentication*
RFC 2961, *RSVP Refresh Overhead Reduction Extensions*
RFC 3097, *RSVP Cryptographic Authentication -- Updated Message Type Value*

RFC 3209, *RSVP-TE: Extensions to RSVP for LSP Tunnels*
RFC 4124, *Protocol Extensions for Support of Diffserv-aware MPLS Traffic Engineering*
RFC 4125, *Maximum Allocation Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering*
RFC 4127, *Russian Dolls Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering*
RFC 4561, *Definition of a Record Route Object (RRO) Node-Id Sub-Object*
RFC 5817, *Graceful Shutdown in MPLS and Generalized MPLS Traffic Engineering Networks*

8.22 Routing Information Protocol (RIP)

RFC 1058, *Routing Information Protocol*
RFC 2080, *RIPng for IPv6*
RFC 2082, *RIP-2 MD5 Authentication*
RFC 2453, *RIP Version 2*

8.23 Segment Routing (SR)

RFC 8287, *Label Switched Path (LSP) Ping/Traceroute for Segment Routing (SR) IGP-Prefix and IGP-Adjacency Segment Identifiers (SIDs) with MPLS Data Planes*
RFC 8426, *Recommendations for RSVP-TE and Segment Routing (SR) Label Switched Path (LSP) Coexistence*
RFC 8476, *Signaling Maximum SID Depth (MSD) Using OSPF – node MSD*
RFC 8491, *Signaling Maximum SID Depth (MSD) Using IS-IS – node MSD*
RFC 8660, *Segment Routing with the MPLS Data Plane*
RFC 8661, *Segment Routing MPLS Interworking with LDP*
RFC 8665, *OSPF Extensions for Segment Routing*
RFC 8667, *IS-IS Extensions for Segment Routing*
RFC 8669, *Segment Routing Prefix Segment Identifier Extensions for BGP*
RFC 9256, *Segment Routing Policy Architecture*
RFC 9350, *IGP Flexible Algorithm*

8.24 Simple Network Management Protocol (SNMP)

draft-blumenthal-aes-usm-04, *The AES Cipher Algorithm in the SNMP's User-based Security Model – CFB128-AES-192 and CFB128-AES-256*
draft-ietf-isis-wg-mib-06, *Management Information Base for Intermediate System to Intermediate System (IS-IS)*
draft-ietf-mpls-ldp-mib-07, *Definitions of Managed Objects for the Multiprotocol Label Switching, Label Distribution Protocol (LDP)*

draft-ietf-mpls-lsr-mib-06, Multiprotocol Label Switching (MPLS) Label Switching Router (LSR) Management Information Base Using SMIv2

draft-ietf-mpls-te-mib-04, Multiprotocol Label Switching (MPLS) Traffic Engineering Management Information Base

draft-ietf-ospf-mib-update-08, OSPF Version 2 Management Information Base

draft-ietf-vrrp-unified-mib-06, Definitions of Managed Objects for the VRRP over IPv4 and IPv6 – IPv6

ESO-CONSORTIUM-MIB revision 200406230000Z, esoConsortiumMIB

IANA-ADDRESS-FAMILY-NUMBERS-MIB revision 200203140000Z, ianaAddressFamilyNumbers

IANAifType-MIB revision 200505270000Z, ianaifType

IANA-RTPROTO-MIB revision 200009260000Z, ianaRtProtoMIB

IEEE8021-CFM-MIB revision 200706100000Z, ieee8021CfmMib

IEEE8021-PAE-MIB revision 200101160000Z, ieee8021paeMIB

IEEE8023-LAG-MIB revision 200006270000Z, lagMIB

RFC 1157, A Simple Network Management Protocol (SNMP)

RFC 1212, Concise MIB Definitions

RFC 1215, A Convention for Defining Traps for use with the SNMP

RFC 1724, RIP Version 2 MIB Extension

RFC 1901, Introduction to Community-based SNMPv2

RFC 2206, RSVP Management Information Base using SMIv2

RFC 2578, Structure of Management Information Version 2 (SMIv2)

RFC 2579, Textual Conventions for SMIv2

RFC 2580, Conformance Statements for SMIv2

RFC 2787, Definitions of Managed Objects for the Virtual Router Redundancy Protocol

RFC 2819, Remote Network Monitoring Management Information Base

RFC 2856, Textual Conventions for Additional High Capacity Data Types

RFC 2863, The Interfaces Group MIB

RFC 2864, The Inverted Stack Table Extension to the Interfaces Group MIB

RFC 2933, Internet Group Management Protocol MIB

RFC 3014, Notification Log MIB

RFC 3273, Remote Network Monitoring Management Information Base for High Capacity Networks

RFC 3410, Introduction and Applicability Statements for Internet Standard Management Framework

RFC 3430, Simple Network Management Protocol (SNMP) over Transmission Control Protocol (TCP) Transport Mapping

RFC 3411, An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks

RFC 3412, Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)

RFC 3413, Simple Network Management Protocol (SNMP) Applications

RFC 3414, *User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)*

RFC 3415, *View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*

RFC 3416, *Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)*

RFC 3417, *Transport Mappings for the Simple Network Management Protocol (SNMP) – SNMP over UDP over IPv4*

RFC 3418, *Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)*

RFC 3419, *Textual Conventions for Transport Addresses*

RFC 3584, *Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework*

RFC 3593, *Textual Conventions for MIB Modules Using Performance History Based on 15 Minute Intervals*

RFC 3635, *Definitions of Managed Objects for the Ethernet-like Interface Types*

RFC 3826, *The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model*

RFC 3877, *Alarm Management Information Base (MIB)*

RFC 4001, *Textual Conventions for Internet Network Addresses*

RFC 4022, *Management Information Base for the Transmission Control Protocol (TCP)*

RFC 4113, *Management Information Base for the User Datagram Protocol (UDP)*

RFC 4273, *Definitions of Managed Objects for BGP-4*

RFC 4292, *IP Forwarding Table MIB*

RFC 4293, *Management Information Base for the Internet Protocol (IP)*

RFC 4878, *Definitions and Managed Objects for Operations, Administration, and Maintenance (OAM) Functions on Ethernet-Like Interfaces*

RFC 7420, *Path Computation Element Communication Protocol (PCEP) Management Information Base (MIB) Module*

RFC 7630, *HMAC-SHA-2 Authentication Protocols in the User-based Security Model (USM) for SNMPv3*

8.25 Timing

RFC 3339, *Date and Time on the Internet: Timestamps*

RFC 5905, *Network Time Protocol Version 4: Protocol and Algorithms Specification*

RFC 8573, *Message Authentication Code for the Network Time Protocol*

8.26 Two-Way Active Measurement Protocol (TWAMP)

RFC 5357, *A Two-Way Active Measurement Protocol (TWAMP) – server, unauthenticated mode*

RFC 5938, *Individual Session Control Feature for the Two-Way Active Measurement Protocol (TWAMP)*

RFC 6038, *Two-Way Active Measurement Protocol (TWAMP) Reflect Octets and Symmetrical Size Features*

RFC 8545, *Well-Known Port Assignments for the One-Way Active Measurement Protocol (OWAMP) and the Two-Way Active Measurement Protocol (TWAMP) – TWAMP*

RFC 8762, *Simple Two-Way Active Measurement Protocol – unauthenticated*

RFC 8972, *Simple Two-Way Active Measurement Protocol Optional Extensions – unauthenticated*

RFC 9503, *Simple Two-Way Active Measurement Protocol (STAMP) Extensions for Segment Routing Networks – excluding Sections 3, 4.1.2 and 4.1.3*

8.27 Virtual Private LAN Service (VPLS)

RFC 4761, *Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling*

RFC 4762, *Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling*

RFC 5501, *Requirements for Multicast Support in Virtual Private LAN Services*

RFC 6074, *Provisioning, Auto-Discovery, and Signaling in Layer 2 Virtual Private Networks (L2VPNs)*

RFC 7117, *Multicast in Virtual Private LAN Service (VPLS)*

8.28 Yet Another Next Generation (YANG)

RFC 6991, *Common YANG Data Types*

RFC 7950, *The YANG 1.1 Data Modeling Language*

RFC 7951, *JSON Encoding of Data Modeled with YANG*

Customer document and product support



Customer documentation

[Customer documentation welcome page](#)



Technical support

[Product support portal](#)



Documentation feedback

[Customer documentation feedback](#)