The SpiderCloud® OS (SCOS) CLI User Guide, Release 5.1

Part number: DOC-SCOS-CLI-08, Rev. 1
Published: January, 2016

Revision History

Revision	Date	Summary of Changes
1	1/6/2016	Initial release for SCOS R5.1

SpiderCloud Wireless
408 East Plumeria Drive
San Jose, CA 95134, USA

http://www.spidercloud.com
Tel: +1 408 567-9165
Email: info@spidercloud.com

Table of Contents

1 Overview

This chapter contains the following sections:

1.1 About this Manual

This guide provides an introduction to the key features and functionalities of the SpiderCloud Operating System (SCOS) Command Line Interface (CLI). It explains the CLI hierarchy, command modes, and command syntax. The guide explains how to interpret the SpiderCloud data model and shows how to use CLI to configure and view data model objects to provision, operate, monitor, and maintain the system. It then explains how to use custom read-only show commands to surveil the system and troubleshoot problems.

This guide is designed to be used in conjunction with the *SCOS NB Data Model Reference Guide* and the *SpiderCloud OS (SCOS) Administrator Guide*. Refer to the data model for details about objects and parameters that comprise the system configuration and operational state. Refer to the administrator guide for information about configuring the software environment and internetworking between the services node and radio node devices.

The primary audience for this guide includes system administrators, network operators, and other personnel responsible for configuring, administrating, and operating the SpiderCloud system. It assumes you have an understanding of the Internet, networking principles, networking configuration, and experience in radio networks.

1.2 Document Conventions

This document uses the following typographical conventions:

- `Monospaced` text indicates CLI input or output. Input is in bold text, output in plain text. For example:

```
show Time NTPServer1
NTPServer1 10.1.11.200;
```

- **Bold text** also indicates a key pressed on a keyboard or other important element. **`Bold monospaced`** text indicates the name of a command or user screen input.
- *Italicized* text indicates a system element that can be configured in the procedure.
- Parameters for input commands are displayed by angle brackets (**<**parameter **>**). For example, **`set IPInterfaceIPAddress <ip_address>`**.

For the sake of brevity, some screen output will be truncated to remove repetitive and non-essential displays and blank lines.

1.3 The SpiderCloud Documentation Set

The SpiderCloud documentation set includes:

- The *SpiderCloud System Description* provides an overview of how the SpiderCloud system fits within an operator's network and in an enterprise, describes key features of the system, and provides specifications for the services and radio nodes.
- The *SpiderCloud Feature Description* provides high-level descriptions of the E-RAN system features, their impact on the product components (services nodes and radio nodes), manageability considerations, and feature benefits.
- The *SpiderCloud OS (SCOS) Administrator Guide* provides procedures for configuring the software environment and internetworking between the services node and radio node devices.
- The *SpiderCloud Services Node Hardware Installation Guide* provides hardware specifications and installation instructions.
- The *SpiderCloud Radio Node Hardware Installation Guide* provides hardware specifications and installation instructions.
- The *E-RAN Deployment Planning Guide* provides information about planning and dimensioning E-RAN systems.
- The *SpiderCloud OS (SCOS) CLI User Guide* provides an introduction to the key features and functionalities of the SpiderCloud Command Line Interface (CLI).
- The *SCOS NB Data Model Reference Guide* provides details about the objects and parameters that comprise the system configuration and operational state.
- The *SpiderCloud OS Faults, Conditions, and Events Reference Guide* provides details about all alarms, conditions, and events in the system.
- The *SpiderCloud System Commissioning Guide* provides information about turning up a SpiderCloud E-RAN with the Local Configuration Interface (LCI) graphical user interface.
- The *Performance Measurements for Small-Cell E-RANs* provides a reference guide to Key Performance Indicators (KPI) that monitor the health and state of the E-RAN system.
- The *E-RAN Troubleshooting Guide* provides information about diagnosing and correcting problems with installing, provisioning, administering, and maintaining SpiderCloud equipment and services.
- The *SpiderNet Management System Installation and Administration Guide* provides information about installing the SpiderNet network management server and client and using it to remotely manage E-RAN deployments.
- The *SpiderCloud Technical Product Description SCSN* provides system specifications for the SpiderCloud services node.
- The *SpiderCloud Technical Product Description SCRN* provides system specifications for the SpiderCloud radio node.
- The *SpiderCloud Time Zone Reference Guide* provides the information required to configure the time zone for SpiderCloud services nodes.
- The *SpiderCloud Call Performance Event Reporting Guide* provides detailed information about call performance events files including the file format, reported events, and event parameters.
- The *SpiderNet NBI Integration Guide* provides information about integrating the SpiderNet network management system into operator's Northbound Interface (NBI) Operations Support Systems (OSSs) to surveil SpiderCloud networks and the SpiderCloud REST API.

2 The SpiderCloud CLI

This chapter contains the following sections:

2.1 CLI Overview

The SpiderCloud Command Line Interface (CLI) is an industry-standard hierarchical text interface for configuring the SpiderCloud services node and its subtended SpiderCloud radio nodes to provide Enterprise Radio Access Network (E-RAN) mobile broadband services. The single-lined commands execute when you press the **Enter** key.

Note

All CLI commands are executed on the services node. Radio nodes are provisioned by the services node.

The SpiderCloud CLI has the following features:

- **Command help:** From any prompt or command, press the **Tab** key or type **?** (question mark) to display the list of valid commands or parameters.

```
admin% set FAPService 1 AccessMgmt?
Possible completions:
  AccessMgmt - Access management configuration
admin% set FAPService 1 AccessMgmt
```

- **Command completion:** From any prompt or command, press the **Tab** key to complete a partially entered command. If there are more than one possible completion, it completes to the point of ambiguity. Press the **Tab** key again to see a list of valid completions. Command completion also applies to other strings such as file and user names.
- **Command memory:** From any prompt, press the ↑ (up arrow) to scroll through the most recently entered commands.
- **Command mobility:** From anywhere on an unexecuted command, use the ← (left arrow) or → (right arrow) to move the cursor on that line for command editing.

2.2 Command Modes

The SpiderCloud CLI has two command modes, each with its own set of commands:

- **Operational Mode**: For monitoring the system and performing basic system administration, such as software upgrades. When you initially log into the services node, your CLI parser is automatically placed into the Operational Mode. A greater-than symbol (>) at the end of the hostname prompt indicates the Operational Mode:

  ```
  admin@sn>
  ```

  Refer to Chapter 5, “Operational Mode Commands.” on page 43 for detailed information about the Operational Mode.

- **Configuration Mode**: For manipulating the system configuration. A percent symbol (%) at the end of the hostname prompt indicates the Configuration Mode. Issue the **`configure`** command to enter the Configuration Mode:

  ```
  admin@sn> configure
  Entering configuration mode private
  admin@sn%
  ```

  Refer to Chapter 4, “Configuration Mode Commands.” on page 33 for detailed information about the Configuration Mode.

2.3 Command Hierarchy

Commands within the CLI are organized in a hierarchy. Related commands are grouped together in sub-levels that may be acted upon by higher level commands. Figure 1 shows an example of the CLI hierarchy. In this example from the Configuration Mode you navigate to the *set* level, then the *FAPService* (the radio node) level, and finally to the *CellConfig* level where you configure the cell parameters.

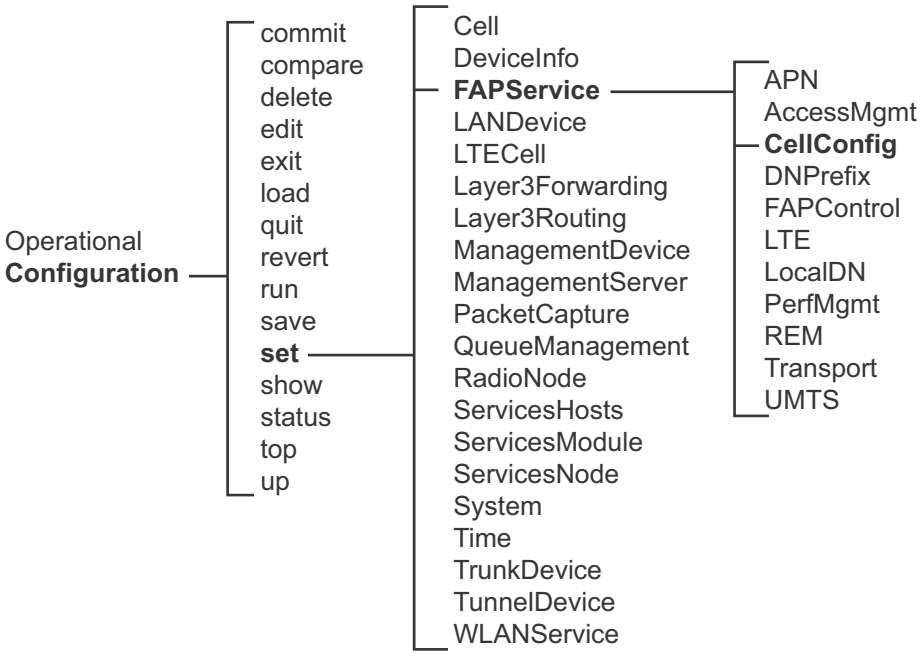

Figure 1 Example of the CLI Hierarchy

The current hierarchical level displays above the command prompt. For example, the root level of the Configuration Mode displays the following hierarchical level indicator and prompt:

```
[edit]
admin@(sn)%
```

By using the edit command, you can focus on a specific level of the hierarchy. Issue the **`edit FAPService <ServiceNumber>`** command to navigate to the *FAPService* level, and the level indicator changes to the following:

```
[edit FAPService 1]
admin@(sn)%
```

Issue the **`edit CellConfig`** command to navigate to the *CellConfig* level, and the level indicator changes to the following:

```
[edit FAPService 1 CellConfig]
admin@(sn)%
```

Recall that you can press the **Tab** key or type **?** (question mark) to display the list of valid commands or parameters.

Issue the **`exit`** command to navigate one level up the hierarchy to the *FAPService* level, and the level indicator returns to the following:

```
[edit FAPService 1]
admin@(sn)%
```

2.4 Refining Show Command Output

Show commands are a valuable method for surveilling the state of the system and troubleshoot problems. However, since the output can be lengthy, you can process or filter the output to target the specific information you are looking for.

Chapter 6, “Show Commands.” on page 65 contains a comprehensive discussion of show commands.

2.4.1 Processing Command Output

Show command output can be verbose, much of which may not be relevant for your specific purpose. You can process the output of a commands such as *show* using an output redirect by entering the **|** (pipe) modifier. CLI commands support the following redirect targets:

Table 1: Command Output Redirects

Redirects	Description
count	Count the number of lines in the output
except	Show only text that does not matches a pattern
find	Search for the first occurrence of a pattern
linnum	Enumerate lines in the output
match	Show only text that matches a pattern
more	Paginate output
nomore	Suppress pagination
save	Save output text to a new or existing file


Note

Text string searching is case sensitive. Enclose text strings containing special characters in quotation marks (“ “).

The following example uses the output redirect to search for equipment that has been administratively placed out of service:

```
show System Event | match ADMIN_DISABLED
2011-01-01T06:36:50.803017Z W EVENT_ADMIN_DISABLED [MOI="LANDevice.2" ]
2011-01-01T06:36:50.716630Z W EVENT_ADMIN_DISABLED [MOI="Cell.30000" ]
2011-01-01T00:01:40.415149Z W EVENT_ADMIN_DISABLED [MOI="LANDevice.2" ]
2011-01-01T00:01:40.327636Z W EVENT_ADMIN_DISABLED [MOI="Cell.30000" ]
2011-06-06T22:20:11.607247Z W EVENT_ADMIN_DISABLED [MOI="LANDevice.2" ]
2011-06-06T22:20:11.525528Z W EVENT_ADMIN_DISABLED [MOI="Cell.30000" ]
2011-06-03T19:07:50.528407Z W EVENT_ADMIN_DISABLED [MOI="LANDevice.2" ]
2011-06-03T19:07:50.445971Z W EVENT_ADMIN_DISABLED [MOI="Cell.30000" ]
```

The following example saves the current configuration to a text file:

```
show configuration | save /configfiles/configfile.cfg
```

The following example displays the primary scrambling codes deployed in the network:

```
show cell | match PrimaryScramblingCode
                        PrimaryScramblingCode           "[ 12 ]";
                        PrimaryScramblingCode           "[ 13 ]";
                        PrimaryScramblingCode           "[ 15 ]";
                        PrimaryScramblingCode           "[ 18 ]";
                        PrimaryScramblingCode           "[ 19 ]";
                        PrimaryScramblingCode           "[ 20 ]";
                        PrimaryScramblingCode           "[ 21 ]";
                        PrimaryScramblingCode           "[ 22 ]";
                        PrimaryScramblingCode           "[ 24 ]";
                        PrimaryScramblingCode           "[ 25 ]";
                        PrimaryScramblingCode           "[ 26 ]";
                        PrimaryScramblingCode           "[ 28 ]";
                        PrimaryScramblingCode           "[ 29 ]";
```

The following example displays session data for a specified International Mobile Subscriber Identity (IMSI):

```
show Session UMTS history | match <imsi>
```

2.4.2 Filtering Output with Regular Expressions

A regular expression is a text string describing a search pattern. Refine searches to target specific output by filtering with regular expressions. Search stings support multiple filters. Output of multiple search filters must pass the filter of each filter.

Regular expressions are case-sensitive. The SpiderCloud CLI supports the following regular expressions:

Table 2: Supported Regular Expressions

Expression	Description
. (period)	Matches any character.
^	Matches the beginning of a string.
$	Matches the end of a string.

Table 2: Supported Regular Expressions *(continued)*

Expression	Description
[abc...]	Character class, which matches any of the characters “abc...” Character ranges are specified by a pair of characters separated by a -.
[^abc...]	Negated character class, which matches any character except “abc...”.
r1 | r2	Alternation. It matches either *r*1 or *r*2.
r1r2	Concatenation. It matches *r*1 and then *r*2.
r+	Matches one or more *r*s.
r*	Matches zero or more *r*s.
r?	Matches zero or one *r*s.
(r)	Grouping. It matches *r*.

The following example searches for UMTS voice sessions that were disconnected due to failure, such as Abnormal Release, and displays a list of events that are both not normal and are voice-related:

```
show Session UMTS History | except Normal\ Release | match Voice
Session   IMSI          D  V  ConnectTime                 RRCState    ConnectCause    Cell   DisconnectTime         DisconnectCause
--------  ------------  -  -  --------------------------  ----------  --------------  -----  ---------------------  ---------------
122048  001010123451354  1  1  2011-08-25T12:55:16.772242Z  Cell_DCH  Voice         14  -                      Still Active
122047  001010123451014  0  1  2011-08-25T12:53:27.355962Z  Cell_DCH  Voice         14  2011-08-25T12:53:30.820811Z  All radiolinks failed
122041  001010123451351  0  1  2011-08-25T12:51:05.993090Z  Cell_DCH  Voice         14  -                      Still Active
122039  001010123456812  0  1  2011-08-25T12:51:02.60933Z   Cell_DCH  Voice         14  -                      Still Active
120156  001010123451264  -  -  -                           -         Voice         14  -                      ServicesNode Rebooted
```

2.5 Logging into the CLI

Once the services node has been configured and brought up with the Local Configuration Interface (LCI) and the IPsec tunnel has successfully been brought up, you can log into the CLI remotely through the Secure SHell (SSH) protocol.

The factory default login is set to the following:

- Username is *admin*
- Password is *admin*

To log into the CLI

Step 1 Step 1 Connect to the services node remotely and access the CLI over the IPsec tunnel using SSH by addressing the administrator account at its IP address. This example uses an IPsec internal tunnel with the IP address *192.1.1.1.*

```
ssh admin@192.1.1.1
admin@192.1.1.1's password: admin
admin connected from 10.1.1.2 using ssh on (sn)
admin@sn>
```

2.6 Configuring CLI Settings

The topics in this section explain how to configure CLI-specific session parameters.

2.6.1 Viewing the Current CLI Settings

Issue the **`show cli`** command from the Operational Mode to display the current CLI settings:

```
show cli
autowizard true;
complete-on-space true;
display-level 99999999;
history 100;
idle-timeout 600;
output {
    file terminal;
}
paginate true;
screen {
    length 50;
    width 80;
}
show {
    defaults false;
}
terminal xterm;
```

2.6.2 Configuring the Display Banner

You can optionally configure a text banner that displays immediately upon user log on to the CLI. The banner can be entered directly into the CLI or uploaded from a text file. The banner displays upon the next user log into the system.

To enter a display banner through the CLI

Step 1 From the Operational Mode, issue the **`request system banner load Terminal`** command and press the *Enter* key.

```
request system banner load Terminal
```

Step 2 Enter the banner text. Carriage returns are permitted. Terminate the input by pressing *CTRL+D*. For example:

```
Welcome to the SpiderCloud CLI!
Press the Tab key or type ? (question mark) to display the list of valid commands
or parameters.
```

To create and load a display banner from a file

Step 1 Create a text file and enter the contents of the display banner. Save the text file.

Step 2 From the Operational Mode, enter the **`request system banner load <FileName>`** command to load the text file. This example uses the file named *banner.txt*.

```
request system banner load banner.txt
```

Viewing the display banner

The banner displays automatically upon the next system log in. For example:

```
ssh admin@10.3.1.18
admin@10.3.1.18's password:
Welcome to the SpiderCloud CLI!
Press the Tab key or type ? (question mark) to display the list of valid commands
or parameters.
admin@>
```

2.6.3 Configuring the Idle Timeout

For security reasons you will automatically be logged out of your CLI session after 600 seconds (ten minutes) of inactivity. This time period is configurable from the Operational Mode on a per-user, per-

session basis by issuing the **set idle-timeout** command and specifying the idle timeout period in seconds. Valid options are from 1 through 8192 (136.5 minutes). Specifying 0 (zero) disables the idle timeout and leaves your session active until the services node reboots, you manually log out, or you are logged out of the session by an administrative user.

The following example sets the idle timeout to *1800* seconds (30 minutes):

```
set idle-timeout 1800
```

2.6.4 Configuring CLI Session Parameters

As a security feature, system administrators can limit the number of CLI sessions per user and total number of all CLI sessions allowed on each services node.

2.6.4.1 Configuring the Number of CLI Sessions per User

An individual user can have multiple concurrent CLI sessions up to the maximum number of total CLI sessions per services node. Use the **set System CLI MaxSessionsPerUser** command from the Configuration Mode to change the maximum number of sessions allowed for each user up to the total number of CLI sessions allowed per services node. The following example sets the maximum number of sessions for each user to *1*.

```
set System CLI MaxSessionsPerUser 1
```

2.6.4.2 Configuring the Total Number of CLI Sessions

By default, the services node supports up to four active CLI administrative user sessions on the services node at one time. If an individual user has multiple concurrent CLI sessions, each count against the total number of sessions. Use the **set System CLI MaxSessions** command from the Configuration Mode to change the maximum number of users allowed within the range of 1 through 4. The following example sets the maximum number of administrative user sessions to *1*.

```
set System CLI MaxSessions 1
```

2.6.5 Viewing CLI Administrator Sessions

The services node supports customizable number active CLI administrative user sessions on the services node at one time. An individual CLI user can have multiple simultaneous sessions as long as the total number of all administrative user sessions does not exceed that number.

Issue the **show users** command from the Operational Mode to view the active CLI users. For example:

```
show users
SID USER  CTX FROM        PROTO LOGIN
 10 admin cli 10.3.254.34 ssh   04:49:42
*8  admin cli 10.1.1.101  ssh   08:33:19
```

Your session is denoted with an ***** (asterisk) before the session ID.

2.6.6 Logging a User Off the System

Use the **request system logout user** command from the Operational Mode to log an administrative user from a CLI session. The following example logs out user *10*:

```
request system logout user 10
```

2.7 User Administration

The SpiderCloud system administrators configure, surveil, and manage the equipment and services in the network. The services node ships with three predefined administrative users: operator administrator, enterprise administrator, and read-only administrator. Each administrative user has its own user group. Table 3 shows the predefined users and their user and group numbers:

Table 3: Predefined Users and Groups

User	CLI Name	User Number	Group Number
Operator administrator	admin	9000	900
Read-only administrator	roadmin	9050	905
Enterprise administrator	N/A	9100	910

The operator administrator can execute all commands. The read-only administrator is restricted to the Operational Mode, and can view configuration, statistical, and log information, and perform a limited number of file management tasks. All read-only administrator tasks are captured in the audit log. Table 4 shows the read-only administrator CLI command permissions:

Table 4: Read Only Admin Command Permissions

Commands Allowed	Commands Not Allowed
exit	configure (cannot enter configuration mode)
file archive	display
file copy	file storage
file delete	request airlink
file get	request clear-debug
file list	request core
file match	request lte
file put	request management-server
file show	request port-mirroring
id	request radionode replace
ping	request scheduled actions
quit	request statistics cell reset
request interface	request statistics delete all
request log bundle	request statistics reset
request log mark	request statistics serviceavailability
request log rotate	request statistics session reset
request log tail	request statistics session rollsnapshot
request message	request statistics syslog
request radionode led	request statistics system reset
request set-debug	request statistics ue

Table 4: Read Only Admin Command Permissions *(continued)*

Commands Allowed	Commands Not Allowed
request statistics cell refresh	request system
request statistics refresh all	request system bootloader update
request statistics session refresh	request system certificate CACert delete
request statistics system refresh	request system database backup
request test mem-dump	request system database restore
request test nmc-dump	request system diagnostics print-mfg-data
request umts debug	request system diagnostics tlv-write
request umts ue	request system
set autowizard	request test add-cell
set complete-on-space	request test detectedextcell
set idle-timeout	request test detectedneighbor
set paginate	request test ip
set show	request test stop-scw
set system password username roadmin	request umts
show	request umts cell
source	request umts core
	request umts rem
	request umts self-config
	show configuration
	test


Note

Wi-FI and WLAN objects are not currently supported.

2.7.1 Enabling the Read-Only Administrator

The read-only administrator has been created by default but must be enabled before the account becomes active.

To enable the read-only administrator

Step 1 From the Configuration Mode, issue the following command to enable the read-only administrator:

```
set System AdminAAA User 9050 Enable true
```

Step 2 Issue the following command to commit the configuration:

```
commit
```

Step 3 Issue the following command to verify the configuration:

```
show System AdminAAA User
User 9050 {
    Enable              true;
    Description         "Operator administrator";
    Username            roadmin;
    GroupID             905;
```

Step 4 From the Operational Mode, issue the following command to set the read-only administrator password.

```
run set system password username roadmin
Enter new password:
Re-enter new password:
```

2.7.2 Editing User Attributes

Users and groups cannot be added or deleted. You can modify user attributes such as description, password, and SNMP community permissions.

To edit user attributes

Step 1 From the Configuration Mode, issue the **`set System AdminAAA User`** command to edit the user attributes. This example sets the SNMP authentication to the SHA protocol, uses the default password of *roadminv3*, and sets the SNMP version to *SNMPv3*.

```
set System AdminAAA User 9000 Enable true SNMPAuthKeySHA roadminv3
SNMPAuthProtocol HMACSHAAuthProtocol SNMPPrivProtocol AESCFB128Protocol
SNMPPrivKeyDES  roadminv3 SNMPVersion [ v3 ]
```

Step 2 Issue the **`show System AdminAAA User`** command to verify the configuration:

```
show System AdminAAA User
User 9000 {
    Enable              true;
    SNMPAuthProtocol    HMACSHAAuthProtocol;
    SNMPAuthKeySHA      $obf$ekkhYPVjAw1TDGA2bg5c;
    SNMPPrivProtocol    AESCFB128Protocol;
    SNMPPrivKeyDES      $obf$xw+9Eo4qEBFPSnwIBElBWhMJ;
    SNMPVersion         "[ v3 ]";
```

2.7.3 Changing User Passwords

Valid user passwords include printable alphanumeric characters. SpiderCloud Wireless recommends passwords of at least eight characters with a mixture of numbers and letters. Note that the output of show commands does not return the actual password. It returns an obfuscated text string.

The *admin* user can change its own password and that of *roadmin*. The *roadmin* user can only change its own password.

Note

User names and passwords are case-sensitive.

To change a user password

Step 1 From the Operational Mode, enter the **`set system password username`** command to change the user password. This example changes the password for the *admin* user.

```
set system password username admin
```

Step 2 Enter and reenter the new password:

```
Enter new password: <NewPassword>
Re-enter new password: <NewPassword>
```

3 Structure of the Data Model

This chapter contains the following sections:

3.1 The SCOS Data Model

The SpiderCloud E-RAN supports a rich data model with thousands of unique objects used to configure and monitor system operation, individual cell status, and user active and historical sessions. The data model expands upon data models defined in TR-096 and TR-198 by the Broadband Forum and 3GPP, and includes numerous extensions for SpiderCloud E-RAN-specific functionalities and topologies.

The data model is protocol agnostic in that it can be accessed through standard management protocols, including TR-069, SNMP, and CLI. XML performance reports can be retrieved through FTP and SCP. A management client application can use the CLI to configure every aspect of the system, query all operational states, and display performance counters. An SNMP manager application can view all operational state and performance parameters.

3.2 Data Object Managed Objects

The system is configured and maintained by manipulating physical and logical managed objects. Table 5 shows the supported managed objects in the SpiderCloud E-RAN (represented by the *System* in the object hierarchy):

Table 5: Managed Objects

Manage Object Types	Managed Object Examples	Valid Options
Cell	Cell.57	1 through 2147483647, 100 total
FAPService	FAPService.1	1
Interface	LANDevice.2.IPInterface.10 ManagementDevice.1.IPInterface.1	1 through 4095, 501 total
LTECell	Cell.72	1 through 2147483647, 100 total
LTE Service	FAPService.1.FAPControl.LTE	LTE
LANDevice	LANDevice.3 ManagementDevice	1 through 10 1
Radio	RadioNode.7.Radio.1	1 through 2
RadioNode	RadioNode.37	1 through 1024, 100 total

Table 5: Managed Objects

Manage Object Types	Managed Object Examples	Valid Options
ServicesNode	ServicesNode.1025	1025
System	System	1
UMTS Service	FAPService.1.FAPControl.UMTS	UMTS

Each services node can support up to 100 radio nodes. Each radio node contains one UMTS or LTE radio, or one UMTS and one LTE radio. Each radio is associated with a single cell.

A LANDevice is the physical Gigabit Ethernet port in the services node. The services node can contain up to 512 logical IP Interfaces which correspond to VLANs.

The services node, radio node, and radio are physical hardware elements in the Radio Access Network (RAN). Cells are logical entities associated with the radios. Figure 2 shows the relationships between managed objects in the SpiderCloud E-RAN. Physical objects are denoted by (p). Logical objects are denoted by (l).

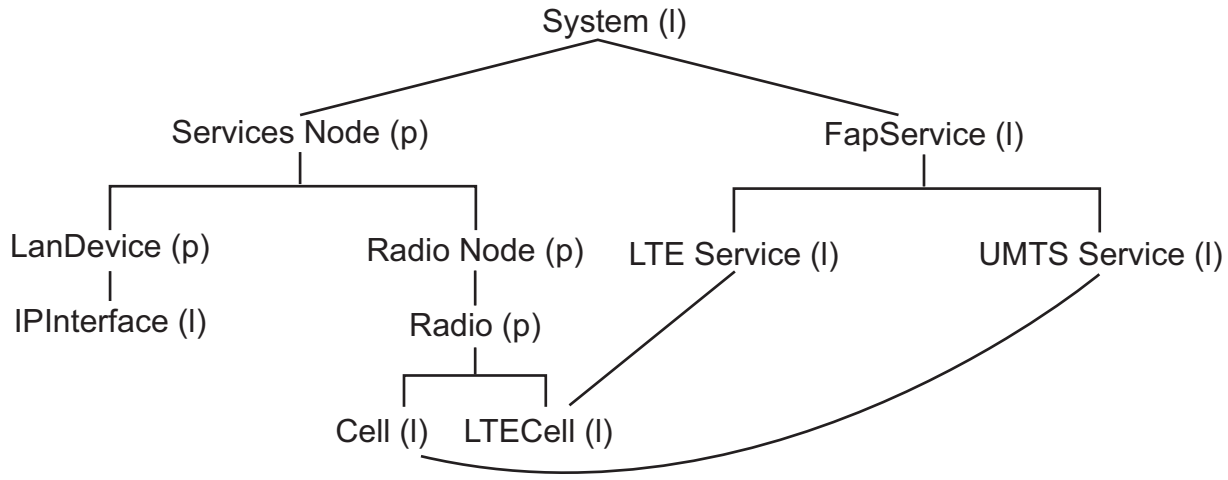

Figure 2 Managed Object Relationships

3.3 The SCOS NB Data Model Reference Guide

The *SCOS NB Data Model Reference Guide* contains the complete set of data model objects and their parameters. It is the exhaustive system reference and should be consulted when the CLI online help does not suffice.

Data model objects are separated by a period (**.**). The lower-case i enclosed in curly braces (*{i}*) is a variable integer that represents an array structure. Every object in the data model is hypertext-linked from the table of contents at the beginning of the data model reference guide. Each of the more than 550 lines of the table of contents:

- represents an object in the SCOS ServicesNode NB data model
- is a link to more information about that object
- reveals the hierarchy by providing the full path to each object

Clicking on a link in the table of contents takes you to that object's description including a table of the direct child parameters of that object. Parameters can be operated upon in the CLI with three types of commands:

- show commands: read-only access to operational state information with a formatted output
- imperative commands: perform an action with a reported result
- configuration commands: perform read-write actions to the configuration hierarchy

The data model reference guide contains more detailed information about parameters and their usage than the brief text displayed in the CLI online help.

For example, the online help for the *PolicyGroupIndex* parameter displays:

```
set FAPService 1 AccessMgmt MemberDetail 1 PolicyGroupIndex?
Possible completions:
  PolicyGroupIndex - Policy group to use for this UE, index of object in
'AccessMgmt PolicyGroup'
```

The data model reference guide entry for this parameter is:

```
FAPService.{i}.AccessMgmt.MemberDetail.{i}.
```

Table 6: PolicyGroupIndex Parameter Information

Parameter	W	Type	Default	Description
PolicyGroupIndex	W	unsignedInt	Mandatory(0)	The value MUST be the instance number of an object in the *.AccessMgmt.PolicyGroup*. table, or 0 if no row is currently referenced. If the referenced object is deleted, the parameter value MUST be set to 0. All members MUST be associated with a policy group. Assigning a set of members (UEs) to the same policy group effectively groups UEs. This allows changes to session parameters for all associated UEs by modifying a single policy group definition. The value MUST be the instance number of an object in the *.AccessMgmt.PolicyGroup.* table, or 0 if no row is currently referenced. If the referenced object is deleted, the parameter value MUST be set to 0. PolicyGroupIndex must not be 0 if Enable is true.

3.3.1 Data Model Parameters

Each item Table 7 represents a parameter and includes the following information:

Table 7: Data Model Parameters

Column	Description
Parameter	Parameter name
Writable	"W" indicates a configuration parameter (read-write) "-" indicates an operational state parameter (read-only)

Table 7: Data Model Parameters *(continued)*

Column	Description
Type	Type of data value: • regex: is a string with pattern restrictions on valid values • enum: is a string with fixed valid values • (min:max): min <= character length of string <= max • [min:max]: min <= integer value <= max • {min:max}: min <= number of items in array <= max • <min:max>: min <= character length of array as comma separated string <= max • (units): units of measurement of an integral type
Default	Parameter default value: • val: val is the default value • <Empty>: "" is the default value and the empty value • Empty(val): val is the default value and the empty value • when parameter value is set to the empty value, the parameter is unspecified and will not be presented upon show in the CLI • Mandatory(val): val indicates the default value of a mandatory parameter • the Description of a mandatory parameter explains when the value of the parameter must be changed to something other than val
Description	Parameter description

Table 8 shows how data model object parameters display in the data model reference guide:

Table 8: Data Model Object Parameter Display

Parameter	W	Type	Default	Description
StandardParameterA	W	enum	WRR	*StandardParameterA* is a configuration parameter as indicated by "**W**". This parameter and its value will be presented when a CLI user executes **`show configuration`** or **`show status OpState`**. **`enum`** indicates this is a string parameter whose possible values are listed below. Enumeration of: • WFQ (Weighted Fair Queueing) • WRR (Weighted Round Robin) • SP (Strict Priority)
StandardParameterB	-	IPAddress	Empty(0.0.0.0)	*StandardParameterB* is an operational state (read-only) parameter as indicated by "**-**". This parameter and its value will be presented when a CLI user executes **`show status OpState`**. If the parameter value is **0.0.0.0**, this indicates that the parameter is unspecified and the parameter will *not* be presented when the CLI user executes **`show status OpState`**.
ExtensionParameter	W	unsignedInt[65535]{1:8}<50>	-	*ExtensionParameter* is a configuration parameter ("**W**") that is also a SCW TR extension: 1. extensions are highlighted by this gray background shading 2. extensions will be presented to an ACS with the appropriate prefix: 3. *X_002448_ExtensionParameter* Default "**-**" indicates there is no default value and that the parameter is mandatory. Parent objects can not exist unless the CLI user provides a value for all mandatory child parameters. The possible range of values for each array item is >= 0 and <= 65535. Array (maximum length as string 50) of *unsignedInt* items (minimum number of items 1, maximum number of items 8).

3.4 Mapping the Data Model to the CLI Hierarchy

As discussed in Command Hierarchy on page 10, data model objects and their related CLI commands are hierarchical or nested. Related commands are grouped together in sublevels below a higher-level command that can act on each command in the sublevel.

Every data model object and parameter can be mapped to the CLI hierarchy:

- In the Operational Mode, use the **`show status OpState`** series of commands to monitor the state of the system as discussed in Section 6.3, Using Show Status OpState on page 85.
- In the Configuration Mode, use the **`set`** and **`show`** commands to monitor the system and provision any configurable parameter.

Figure 3 shows a simple example of the CLI hierarchy. In this example, from the Configuration Mode, you navigate to the *set* level, then the *RadioNode* (the radio node) level where you assign it a number, then configure one or more parameters.

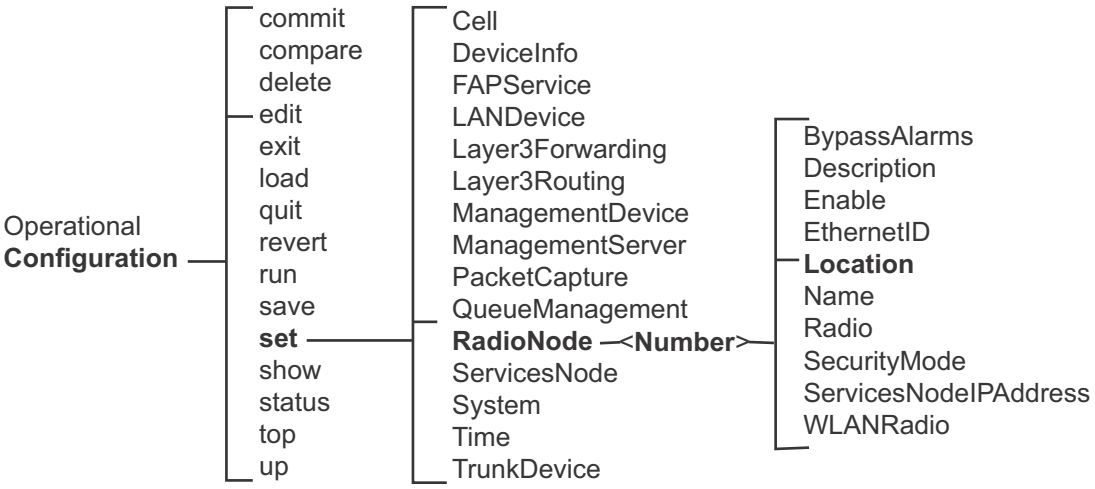

Figure 3 RadioNode Object Hierarchy

Note

Wi-FI and WLAN objects are not currently supported.

The CLI command for setting the radio node location attributes is **`set RadioNode <Number> Location`**. Assign the radio node number *44*. Using the **Tab** key for command completion, the system responds with the available parameters:

```
set RadioNode 44 Location
Possible completions:
  Altitude                        - Specifies the altitude in meters
  Latitude                        - Specifies the latitude in deg x 1M
  Longitude                       - Specifies the longitude in deg x 1M
[Output truncated]
```

The data model equivalent to **`set RadioNode <Number> Location`** is:

`RadioNode.{i}.Location.`

The data model displays the SpiderCloud proprietary parameters for **`Location`** (table edited):

Table 9: RadioNode Location Parameters

Parameter	W	Type	Default	Description
Altitude	W	int[ranges]	Mandatory (2147483647)	Specifies the altitude in meters. positive value signifies height (direction Up) negative value signifies depth (direction Down) Possible values: • -32767 to 32767 • 2147483647 Altitude must not be 2147483647 if *GeographicalAreaFormat* is *PointAndAltitude*.
Latitude	W	int[ranges]	Mandatory (2147483647)	Specifies the latitude in degrees multiplied by 1 million (*deg x 1M*). • positive value signifies a direction North of the equator • negative value signifies a direction South of the equator Range is from: 90{{degrees}}00.00' South (-90,000,000) to 90{{degrees}}00.00' North (90,000,000). Example value: 13,323,833 (13{{degrees}}19.43' N, derived as (13*1,000,000)+((19.43*1,000,000)/60). Example value: -50,000,000 (50{{degrees}}0.00' S) Possible values: • -90000000 to 90000000 • 2147483647 Latitude must not be 2147483647 if *GeographicalAreaFormat* is Point or *PointWithUncertainty* or *PointAndAltitude* or *PointAndAltitudeWithUncertainty*.

Table 9: RadioNode Location Parameters *(continued)*

Parameter	W	Type	Default	Description
Longitude	W	int[ranges]	Mandatory (2147483647)	Specifies the longitude in degrees multiplied by 1 million (*deg x 1M*). • positive value signifies a direction East of the prime meridian • negative value signifies a direction West of the prime meridian Range is from: 180{{degrees}}00.00' West (-180,000,000) to 180{{degrees}}00.00' East (180,000,000). Example value: 13,323,833 (13{{degrees}}19.43' E, derived as (13*1,000,000)+((19.43*1,000,000)/60). Example value: -50,000,000 (50{{degrees}}0.00' W) Possible values: • -180000000 to 180000000 • 2147483647 Longitude must not be 2147483647 if *GeographicalAreaFormat* is Point or *PointWithUncertainty* or *PointAndAltitude* or

3.5 Using the CLI to Configure Data Model Attributes

Once you understand the relationship between the data model and the CLI hierarchy, you can use the *SCOS NB Data Model Reference Guide* to assist with converting the data model objects and their parameters into CLI commands to provision and monitor the system.

3.5.1 Typical UMTS Configuration

This section provides a detailed example of a more complex configuration than the example used in Section 3.4, Mapping the Data Model to the CLI Hierarchy on page 26. It will set the nested objects and parameters for the radio frequency of a cell along the following data model path:

`Cell.{i}.CellConfig.UMTS.RAN.FDDFAP.RF`.

This CLI example sets the following hierarchical UMTS cell properties in sequence:

- the cell number
- the cell description, cell name, radio node, and radio numbers; and then enables the cell
- cell parameters for the radio, radio network, and radio services
- cell-level configuration for air interface cell related properties and radio frequency parameters

Using the **`set`** command as an example from the Configuration Mode, the following options are available at the first (cell) level:

```
set Cell
Possible completions:
  <index:unsignedInt, <= 4294967294, >= 1>  1  2  3  4  5  6  7  8
```

At this level, the only option is to define the cell number. Valid options are numbers from 1 through 4294967294. We will define the properties of cell 55:

```
set cell 55
Possible completions:
  AccessMgmt  - Access management configuration
  CellConfig  - Cell specific configuration parameters
  Description - Description of this Cell
  Enable      - Enables or disables this Cell
  Name        - Name of this Cell
  Radio       - The index of the Radio associated with this Cell
  RadioNode   - The index of the RadioNode associated with this Cell
```

At this level you can assign the cell a description and a name, and assign its radio node and radio a number as well as enable and disable it. These actions can be performed in a single command:

```
set Cell 55 Description Kitchen Name Cell_55 Enable true Radio 1 RadioNode 55
```

Use the **`show Cell 55`** command to validate the entry:

```
show cell 55
Enable      true;
Name        Cell_55;
Description Kitchen;
RadioNode   55;
Radio       1;
```

At this level you can also navigate into one of two nested hierarchies, *AccessMgmt* for parameters related to user access to the system, or to *CellConfig* for parameters related to configuring the Femto Access Point (FAP) which are the radio services. We will provision the radio services:

```
set Cell 55 CellConfig UMTS RAN
Possible completions:
  CellID - Cell Identity
  FDDFAP - Parameters relating to the cell-level configuration for an FDD mode
           cell
```

The only completion for **`CellConfig`** is **`UMTS`** for properties related to the radio. The only completion for **`UMTS`** is **`RAN`**, for properties related to the radio network. At this level you can assign the cell an ID or assign parameters relating to the cell-level configuration for air interface cell related properties (FDDFAP). We will define the FDDFAP properties:

```
set Cell 55 CellConfig UMTS RAN FDDFAP RF
Possible completions:
  MaxFAPTxPower                    - Value or range of maximum cell Transmit
                                     power allowed for the cell
  MaxFAPTxPowerInUseAlarmReference - A known good reference value for
                                     MaxFAPTxPowerInUse
  MaxFAPTxPowerLockEnable          - Locks or unlocks MaxFAPTxPower used by
                                     the cell
  Mode                             - Intended cell mode - scan or operational
                                     mode
  PrimaryScramblingCode            - Primary DL Scrambling Code value or set
                                     allowed for the cell
  PrimaryScramblingCodeConfigured  - Configured Primary DL Scrambling Code for
                                     the cell
  PrimaryScramblingCodeLockEnable  - Locks or unlocks Primary DL Scrambling
                                     Code in use by the cell
  RFLockEnable                     - Locks or unlocks the Primary Scrambling
                                     Code, MaxFAPTxPower and neighbor list
                                     configured for the cell
  ResetMode                        - Deprecated parameter, use Mode instead
                                     (DEPRECATED)
  UARFCNDL                         - DL UTRA Absolute Radio Frequency Channel
                                     Number (UARFCN) value or set allowed for
                                     the cell
  UseSelfConfigAlternatePSC        - When NeighborListSelfConfigEnable and
                                     UseSelfConfigAlternatePSC are both true,
                                     this cell is assigned a PSC from
```

```
                                    SelfConfigAlternatePSC rather than
                                    PrimaryScramblingCode
```

The only completion for **FFDAP** is **RF** where you can define a number of radio frequency parameters with a single command:

```
set Cell 55 CellConfig UMTS RAN FDDFAP RF MaxFAPTxPower 90 Mode UMTSNodeB
UARFCNDL [ 1937 ] UseSelfConfigAlternatePSC false
```

Use the **show Cell 55** command to validate all cell entries:

```
show Cell 55
Enable      true;
Name        Cell_55;
Description Kitchen;
RadioNode   55;
Radio       1;
CellConfig {
    UMTS {
        RAN {
            FDDFAP {
                RF {
                    UARFCNDL                          "[ 1937 ]";
                    PrimaryScramblingCode             "[ 0..511 ]";
                    MaxFAPTxPower                     90;
                    UseSelfConfigAlternatePSC         false;
                    ResetMode                         true;
                    Mode                              UMTSNodeB;
                    MaxFAPTxPowerInUseAlarmReference 65535;
                }
            }
        }
    }
}
```

3.5.1 Typical LTE Configuration

This section provides a detailed example of a more complex configuration than the example used in Section 3.4, Mapping the Data Model to the CLI Hierarchy on page 26. It will set the nested objects and parameters for the radio frequency of a cell along the following data model path:

LTECell.{i}.CellConfig.LTE.RAN.RF.

This CLI example sets the following hierarchical LTE cell properties in sequence:

- the cell number
- the cell description, cell name, radio node, and radio numbers; and then enables the cell
- cell parameters for the radio, radio network, and radio services
- cell-level configuration for air interface cell related properties and radio frequency parameters

Using the **set** command as an example from the Configuration Mode, the following options are available at the first (cell) level:

```
set LTECell
Possible completions:
  <index:unsignedInt, <= 2147483647, >= 1>
```

At this level, the only option is to define the cell number. Valid options are numbers from 1 through 2147483647. We will define the properties of cell 56

```
set LTECell 55
Possible completions:
  BypassAlarms - A list of alarms to bypass
  CellConfig   - Cell specific configuration parameters
  Description  - Description of this LTECell
```

```
  Enable       - Enables or disables this LTECell
  LocationType - Description of the location where this LTECell is installed
  Name         - Name of this LTECell
  Radio        - The index of the Radio associated with this LTECell
  RadioNode    - The index of the RadioNode associated with this LTECell
```

At this level you can assign the cell a description and a name, and assign its radio node and radio a number as well as enable and disable it. These actions can be performed in a single command:

`set LTECell 56 Description Conference Name 56 Enable true Radio 1 RadioNode 56`

Use the **`show LTECell 56`** command to validate the entry:

```
show LTECell 56
Enable       true;
Name         56;
Description  Conference;
RadioNode    56;
Radio        1;
```

3.5.2 Short Configuration

In many cases it is not necessary to provision every possible attribute. For example, you may simply want to enable a cell:

`set Cell 66 Enable true`

Use the **`show Cell 66 Enable`** command to validate the entry:

```
show Cell 66 Enable
Enable true;
```

`set LTECell 67 Enable true`

Use the show **`LTECell 67`** Enable command to validate the entry:

```
show LTECell 67 Enable
Enable true;
```

4 Configuration Mode Commands

The Configuration Mode provides access to all read-write parameters of the SCOS data model. Use the Configuration Mode to provision the SpiderCloud system and make changes to the system configuration. A percent symbol (%) at the end of the hostname prompt indicates the Configuration Mode. Issue the **`configure`** command from the Operational Mode to enter the Configuration Mode:

```
admin@sn> configure
Entering configuration mode private
admin@sn%
```

This chapter contains the following sections:

The Configuration Mode has the following top-level commands:

- **commit:** Commit current set of changes
- **compare:** Show configuration differences
- **delete:** Delete a data element
- **edit:** Edit a sub-element
- **exit:** Exit from this level
- **load:** Load configuration from an ASCII file
- **quit:** Exit from this level
- **revert:** Discard any outstanding edits
- **run:** Run an Operational Mode command
- **save:** Save configuration to an ASCII file
- **set:** Set a parameter
- **show:** Show a parameter
- **status:** Display users currently editing the configuration
- **top:** Exit to top level and optionally run command
- **up:** Exit one level of configuration

4.1 Working in the Configuration Mode

4.1.1 Entering and Navigating the Configuration Mode

- From the Operational Mode, issue the **configure** command to enter the Configuration Mode:

```
admin@sn> configure
Entering configuration mode private

[edit]
admin@%
```

- Use the **edit** command to move to a hierarchy level. For example, the command **edit RadioNode 1 Radio 1** enters the [edit RadioNode 1 Radio 1] hierarchy level.

  The **edit** command changes hierarchy levels in the CLI and operates like the **cd** command in UNIX. (Issuing CD in that environment moves you to a new directory level.)

```
[edit]
admin@sn% edit RadioNode 1 Radio 1

[edit RadioNode 1 Radio 1]
admin@sn%
```

- Use the **set** command to configure parameters at the current hierarchy level.

```
[edit RadioNode 1 Radio 1]
admin@sn% set Enable false
```

- Use the **exit** command to navigate to the previous hierarchy level.

```
[edit RadioNode 1 Radio 1]
admin@sn% exit

[edit]
admin@sn%
```

4.1.2 Entering Configuration Mode Commands

You can enter Configuration Mode commands individually or with one extended compound command. The following examples show how to define the two NTP servers for the services node using the **set** command. The **set** command modifies existing configuration attributes or creates them if they do not previously exist. Bolded text indicates user input.

- Entering the commands and parameters individually:

```
[edit]
admin@(sn)% set Time
[ok][2011-01-04 21:35:39]

[edit Time]
admin@(sn)% set NTPServer1 10.202.1.1
[ok][2011-01-04 21:36:31]

admin@(sn)% exit
[ok][2011-01-04 21:36:52]

admin@(sn)% set NTPServer2 10.202.2.2
[ok][2011-01-04 21:37:25]
```

```
[edit Time NTPServer2]
admin@(sn)% exit
[ok][2011-01-04 21:37:33]
```

- Entering a single, compound command with the associated parameters:

```
admin@(sn)% edit Time NTPServer1 10.202.1.1 NTPServer2 10.202.2.2
```

4.1.3 Operational Mode Commands in the Configuration Mode

Use the **run** command to issue an Operational Mode command in the Configuration Mode. The benefit of using the **run** command is that you can issue an Operational Mode command in the Configuration Mode. Therefore, you avoid having to commit a candidate configuration or toggle between the two modes to issue Operational Mode commands.

```
run show ServicesNode Time
ServicesNode 1025:
    CurrentTime: 2013-01-04T16:49:23Z
    ArriveTime:  2013-01-04T15:39:29Z
    UpTime:      01:09:54
```

4.1.4 Using Show Commands as Provisioning Aids

In the Configuration Mode, the **show** commands display the command hierarchy required to provision the system. For example, issuing the **show LANDevice** command returns the following:

```
show LANDevice
LANHostConfigManagement {
    IPInterface 1 {
        Enable                true;
        IPInterfaceIPAddress  10.1.192.10;
        IPInterfaceSubnetMask 255.255.255.0;
    }
    IPInterface 2 {
        Enable                true;
        IPInterfaceIPAddress  10.1.192.3;
        IPInterfaceSubnetMask 255.255.255.0;
        VLANID                2;
    }
}
LANEthernetInterfaceConfig 1 {
    Enable     true;
    MaxBitRate 1000;
    DuplexMode Full;
}
```

The same output displays from the Operational Mode by entering a **show configuration** command**.** In the example above, issue the **show configuration LANDevice** command.

From the output of the **show LANDevice** command above you can deduce the following:

- The *LANHostConfigManagement* object is the parent of *IPInterface* and *LANEthernetInterfaceConfig*. Therefore, *IPInterface* and *LANEthernetInterfaceConfig* are configured on the *LANHostConfigManagement* hierarchical level.
- *IPInterface* is the parent of *Enable*, *IPInterfaceIPAddress*, and *IPInterfaceSubnetMask*. Therefore, *Enable*, *IPInterfaceIPAddress*, and *IPInterfaceSubnetMask* are configured on the *LANHostConfigManagement* | *IPInterface* hierarchical level.

- The *LANEthernetInterfaceConfig* object is the parent of *Enable*, *MaxBitRate*, and *DuplexMode*. Therefore, are configured on the *LANHostConfigManagement* | *LANEthernetInterfaceConfig* level.

The *LANHostConfigManagement* object is a child of the *LANDevice* object in the Configuration Mode. With this understanding, use the following commands to create the configuration that produces the example of the **`show LANDevice`** command above:

To create the sample configuration

Step 1 From the Configuration Mode, issue the following commands to configure the LAN host:

```
admin@(sn)% set LANDevice 1 LANHostConfigManagement
[ok][2011-01-05 01:03:02]

[edit LANDevice 1 LANHostConfigManagement]
admin@(sn)% edit IPInterface 1 Enable true IPInterfaceIPAddress 10.1.192.3
IPInterfaceSubnetMask 255.255.255.0
[ok][2011-01-05 01:10:39]

admin@(sn)% exit
[ok][2011-01-05 01:10:49]

[edit LANDevice 1 LANHostConfigManagement]
admin@(sn)% set LANEthernetInterfaceConfig 1 Enable true
[ok][2011-01-05 01:12:17]

[edit LANDevice 1 LANEthernetInterfaceConfig 1 Enable]
admin@(sn)% exit
[ok][2011-01-05 01:13:19]

[edit LANDevice 1]
admin@(sn)% commit
Commit complete.
```

4.1.5 Deleting Objects

Use the **`delete`** command to remove SCOS objects and their parameters from the candidate configuration. When you delete a parameter, the parameter value is reset to the default value in the candidate configuration. The following example deletes cell *66*:

```
delete Cell 66
```

Issue the **`show cell 66`** command to validate the configuration:

```
show cell 66
---------------------------^
syntax error: unknown element
```

4.2 Command Mode Show Commands

Unlike the custom show commands in the Operational Mode, Configuration Mode show commands follow the outline of the data model. Any object in the current system configuration can be viewed with a Configuration Mode show command by following the data model as described in Section 3.4, Mapping the Data Model to the CLI Hierarchy on page 26.

Show command output can be verbose. Consider processing or filtering output as discussed in Section 2.4, Refining Show Command Output on page 11, or making the commands very specific by incorporating more objects.

4.2.1 No Filter

The following example shows the truncated output of a **`show cell`** command with no filter. The output is 217 lines.

```
show Cell
Cell 1 {
    Enable      true;
    Name        near-bathroom;
    RadioNode   1;
    Radio       1;
    AccessMgmt {
        AdmissionControl {
            MobilityLinkReservation 3;
        }
    }
    CellConfig {
        UMTS {
            RAN {
                CellID 65601537;
                FDDFAP {
                    RF {
                        UARFCNDL                        "[ 437 ]";
                        PrimaryScramblingCode           "[ 100..150 ]";
                        MaxFAPTxPower                   -100..200;
                        UseSelfConfigAlternatePSC       false;
                        Mode                            UMTSNodeB;
                    }
                }
            }
        }
    }
}
Cell 2 {

[output truncated]
```

4.2.2 One-Level Filter

The following example shows the output of a one-level **`show cell`** command filter. The output is filtered for cell *1* information only. The output is 26 lines.

```
show cell 1
Enable      true;
Name        near-bathroom;
RadioNode   1;
Radio       1;
AccessMgmt {
    AdmissionControl {
        MobilityLinkReservation 3;
    }
}
CellConfig {
```

```
    UMTS {
        RAN {
            CellID 65601537;
            FDDFAP {
                RF {
                    UARFCNDL                        "[ 437 ]";
                    PrimaryScramblingCode           "[ 100..150 ]";
                    MaxFAPTxPower                   -100..200;
                    UseSelfConfigAlternatePSC       false;
                    Mode                            UMTSNodeB;
                }
            }
        }
    }
}
```

4.2.3 Two-Level Filter

The following example shows the output of a one-level **`show cell`** command filter. The output is filtered for cell *1*, then again for *CellConfig UMTS* (*UMTS* is the only option under *CellConfig*). The output is 13 lines.

```
show Cell 1 CellConfig UMTS
RAN {
    CellID 65601537;
    FDDFAP {
        RF {
            UARFCNDL                        "[ 437 ]";
            PrimaryScramblingCode           "[ 100..150 ]";
            MaxFAPTxPower                   -100..200;
            UseSelfConfigAlternatePSC       false;
            Mode                            UMTSNodeB;
        }
    }
}
```

4.3 Managing the Configuration

Managing the configuration involves viewing the running and candidate configurations, editing the configuration file, saving the running configuration, and loading and merging configuration files. Refer to Section 4.3.8, The Candidate Configuration and the Commit Command on page 42 for information about the difference between the running and candidate configurations.

This section contains the following topics:

4.3.1 Displaying the Running Configuration

You can display the running configuration from the Configuration Mode or the Operational Mode:

- To display the running configuration in the Operational Mode, use the **`show configuration`** command.
- To display the running configuration in the Configuration Mode, issue the **`run show configuration`** command, as shown below:

```
run show configuration
FAPService 1 {
    FAPControl {
        UMTS {
            SelfConfig {
                NeighborListSelfConfigEnable   true;
                MeasIMSIList                   123456789101001;
                MeasLoadingFactor              70;
                FAPCoverageTargetMinBase       -900;
                FAPCoverageTargetValue1        50;
                FAPCoverageTargetValue2        -30;
                FAPCoverageTargetAdditionDelta 30;
                NumMeasUe                      101;
                NumPtsThresh                   101;
                NumImsiThresh                  101;
                NumValidRSThresh               101;
                rMeasDiscard                   101;
                AutoProvisionEnable            true;
            }
            Gateway {
                SecGWServer1 10.1.30.104;
                CNProtocol   Iu/IP;
            }
        }
    }
[output truncated]
```

Note that the output of the **`show configuration`** command is quite verbose. For a more targeted response, add the **`match`** parameter as illustrated below:

```
show configuration | match FACHInactivityTimer
                        FACHInactivityTimer         0;
```

Refer to Section 2.4.1, Processing Command Output on page 11 for more information about filtering show command output.

4.3.2 Displaying the Candidate Configuration

Use the **show** command in the Configuration Mode to display the candidate configuration (the configuration with your edit changes included).

To display the candidate configuration

Step 1 From the Configuration Mode, issue the command to modify the running configuration. This example sets the CN protocol to *Iu/IP*.

```
set FAPService 1 FAPControl UMTS Gateway CNProtocol Iu/IP
```

Step 2 Issue the **show** command to display your changes and other settings for that hierarchical level. In this example, Step 1 set the *CNProtocol* parameter, so the **show** command below extends down to the **`show FAPService <ServiceNumber> FAPControl UMTS Gateway CNProtocol`** level.

```
show FAPService 1 FAPControl UMTS Gateway CNProtocol
```

```
CNProtocol Iu/IP;
```

The following example extends the **show** command to the **show FAPService <ServiceNumber> FAPControl UMTS Gateway** level to display the candidate configuration changes to the UMTS gateway hierarchy:

```
show FAPService 1 FAPControl UMTS Gateway
SecGWServer1 10.1.30.104;
CNProtocol   Iu/IP;
FAPGWPort    29169;
```

4.3.3 Displaying Changes in the Candidate Configuration

To display outstanding changes in the candidate configuration before committing them, issue the **compare running** command in the Configuration Mode. New statements to be added to the configuration are flagged with a plus sign (+). Removed statements are flagged with a minus sign (-).

```
compare running
...
  Cell 1 {
-    Enable true;
+    Enable false;
      RadioNode 1;
      Radio 1;
...
  }
+Cell 2 {
+    Enable true;
+    RadioNode 2;
+    Radio 1;
+    CellConfig {
+        UMTS {
+            RAN {
+                FDDFAP {
+                    MobilityLinkReservation 3;
+                    RF {
+                        UARFCNDL "[ 10700 ]";
+                        PrimaryScramblingCode "[ 0 ]";
+                        MaxFAPTxPower 0;
+                    }
+                }
+            }
+        }
+    }
+}

...

  RadioNode 1 {
-    Enable true;
+    Enable false;
      EthernetID 00:00:00:aa:aa:cc;
      SecurityMode open;
      Radio 1 {
          Enable true;
          Band umts-band-I;
  }
+RadioNode 2 {
+    Name "";
+    Description "";
+    Enable true;
+    EthernetID 00:00:00:aa:aa:bb;
+    SecurityMode open;
```

```
+    ServingController 0.0.0.0;
+    Radio 1 {
+        Enable true;
+        Band umts-band-I;
+    }
+}
[output truncated]
```

4.3.4 Discarding Edits

In the Configuration Mode, you can discard candidate configuration edits in one of two ways:

- Use the **`exit`** command to leave the Configuration Mode before executing the **`commit`** command. Exiting the Configuration Mode without committing changes discards all session edits.
- Use the **`revert`** command.

4.3.5 Saving the Running Configuration to a File

You can save the current running configuration to a text file on the services node.

To save the running configuration

Step 1 From the Configuration Mode, issue the **`save <filename.cfg>`** command. This example saves the running configuration to a file named *config-2011-08-10.cfg*.

```
save config-2011-08-10.cfg
```

Step 2 Issue the **`run file list`** command to verify that the file was successfully saved:

```
run file list
cell-insert.txt
cell-test.txt
config-2011-08-10.cfg
error_incidents/
fg_200_list.txt
[output truncated]
```

Step 3 (Optional) Issue the **`run file list Detail`** command to verify the file timestamp:

```
run file list Detail
drwx------    2      4096 Aug 31 21:22 .ssh/
-rw-r--r--    1      2496 Aug 16 06:32 Cert-SCW_CA1-Self.pem
-rw-r--r--    1      2468 Aug 16 06:33 Cert-db212s6.int.spidercloud.com-SCW_CA1.pem
-rw-r--r--    1      7593 Aug 16 06:31 SpiderCloud-scsn-Cert.pem
-rw-r--r--    1     18568 Aug 13 22:14 access_points_radios_insert_0000128.txt
-rw-r--r--    1      6136 Aug 13 22:14 access_points_radios_iud_0000010.txt
-rw-r--r--    1     35656 Sep  9 00:59 bc_add_cell_test_config
-rw-r--r--    1       130 Aug 13 22:14 cell-delete.txt
-rw-r--r--    1       217 Aug 13 22:14 cell-insert.txt
-rw-r--r--    1     17782 Aug 10 23:19 config-2011-08-10.cfg
```

4.3.6 Backing Up the Running Configuration

After saving the running configuration to a file, you can back it up to an external device. Issue the **`file put <local_path> <destination-url>`** command from the Operational Mode using the Secure Copy Protocol (SCP), File Transfer Protocol (FTP), or Trivial File Transfer Protocol (TFTP). If you do not specify the device password, you will be prompted for it. The password you enter is not echoed back to the terminal.

The following command backs up the file *backup.cfg* to the server with the IP address *10.20.10.1*. Note that this example renames the *backup.cfg* file using the SCP to add the date *08/01/2011*.

```
file put backup.cfg scp://admin@10.20.10.1/backup.08.01.2011.cfg
```

```
admin@10.20.10.1's password:
```

Enter the password to the remote server.

4.3.7 Loading and Merging a Configuration File

Issue the **`load merge <filename.cfg>`** command from the Configuration Mode to load a previously saved configuration file onto the services node and merge this with your current candidate configuration. It will overwrite the existing provisioning for those parameters whose values are different in the new configuration file.

Parameters that are part of the original configuration but not of the new one are maintained. Parameters that are in the new configuration but not in the original one are added to the merged configuration.

Note

If the configuration file was saved from another services node, first use a text editor and change the IP address (in the *LANDevice 1* section) of the other services node to the one of the node you are importing it to.

The following example loads the file named *config-2011-08-10.cfg* and then commits it, making it the running configuration.

```
load merge config-2011-08-10.cfg
commit
```

4.3.8 The Candidate Configuration and the Commit Command

In the Configuration Mode, commands entered and text returned on the screen apply to the candidate configuration. The candidate configuration is not applied to the active system until the **`commit`** command is entered and validated. At this point it becomes the active configuration. Use the candidate configuration to build and modify the system without interfering with actual network operations.

5 Operational Mode Commands

This chapter contains the following sections:

Use the Operational Mode to monitor the system and perform basic system administration such as software upgrades, file management, and rebooting the system. When you initially log into the services node, your CLI parser is automatically placed into the Operational Mode. A greater-than symbol (**>**) at the end of the hostname prompt indicates the Operational Mode:

```
admin@sn>
```

The Operational Mode contains the following top-level commands:

- **configure:** Enter the Configuration Mode to manipulate software configuration information.
- **display: Display operational state.**
- **exit:** Exit the CLI management session.
- **fetch:** Retrieve object attributes. Refer to Section 5.3, The Fetch Command on page 54.
- **file:** Perform file operations. Refer to Section 5.1, Managing Files on page 44
- **id:** Show user ID information.
- **ping:** Send ICMP ECHO_REQUEST to network host. Refer to Section 5.5, Additional Utilities on page 62.
- **ping6:** Send ND ECHO_REQUEST to network host to discover neighbors in an IPv6 network.
- **quit:** Exit the CLI management session.
- **request:** Make system-level requests. Refer to Section 5.4, Request Commands on page 56.
- **set:** Set CLI properties.
- **show:** Show information about the system. Refer to Chapter 6, “Show Commands” on page 65.
- **source:** File to source.
- **test:** Test the configuration. Refer to Section 5.5, Additional Utilities on page 62.
- **traceroute:** Print the route packets take to the network host.

This chapter contains the following sections:

5.1 Managing Files

The following sections have information about managing files on the services node:

5.1.1 Using the *file list* Command

Core dumps and error bundles for both the services node and radio nodes are stored in the services node under the *error_incidents* directory. Use the **`file list <directory_to_list>`** command from the Operational Mode to display the contents of that directory:

```
file list error_incidents/
```

The file name syntax includes the process name that crashed, the process ID that was running at the time of the crash, and the date and time when the core dump was generated:

```
<SN | RN>-<node id>.<process name>.<pid>.<date YYMMDD>.<time HH_MM_SS>.tgz
```

For example:

```
SN-1025.addressd.17863.091007.11_24_58.tgz
```

Component log files are stored under the *logfiles* directory. The system maintains a logging history of 10 files with a size limitation of a 50MB for each log file. Issue the **`file list logfiles/`** command in the Operational Mode to display the contents of a logfiles directory:

```
file list logfiles/
audit.log
confd.log
db_logfile-2014-10-01_225251.log
db_logfile-2014-10-01_230355.log
db_startup.log
debug.log
debug.log.ac-2015.141001.22_52_46
debug_session.log
debug_session.log.ac-2015.141001.22_52_46
devel.log
event.log
event.log.01
messages
wtmp

[output truncated]
```

Use the **`file list Detail`** command to display detailed information about files in the root directory:

```
file list Detail
drwx------    2      4096 Aug 17 22:36 .ssh/
-rw-r--r--    1       343 Aug 31 21:16 readme
drwxr-xr-x    2      4096 Aug 30 07:11 error_incidents/
-rw-r--r--    1      2494 Sep  1 11:29 filecopy-2011-09-01_112940
-rw-r--r--    1      2494 Sep  1 11:29 filecopy-2011-09-01_112942
-rw-r--r--    1      2415 Sep  1 11:29 filecopy-2011-09-01_112944
```

Refer to Section 2.4.1, Processing Command Output on page 11 and Section 2.4.2, Filtering Output with Regular Expressions on page 12 for information about refining command output.

Use the **`file list Detail <directory_to_list/>`** command to display detailed information about the files in another directory:

```
file list Detail logfiles/
-rw-r--r--    1   2797079 Aug 29 21:15 audit.log
-rw-r--r--    1     90910 Aug 27 10:17 confd.log
-rw-------    1    244941 Aug 26 10:13 db_logfile-2011-08-26_070840.log
-rw-------    1      6507 Aug 26 14:53 db_logfile-2011-08-26_101410.log
-rw-------    1      6507 Aug 26 18:44 db_logfile-2011-08-26_145351.log
-rw-------    1      8843 Aug 26 18:48 db_logfile-2011-08-26_184441.log
-rw-------    1       465 Aug 26 19:35 db_logfile-2011-08-26_184840.log
[output truncated]
```

5.1.2 Using the *file show* Command

Issue the **`file show <file_to_show>`** command from the Operational Mode to display the contents of a file saved on the services node. Refer to Section 2.4.1, Processing Command Output on page 11 and Section 2.4.2, Filtering Output with Regular Expressions on page 12 for information about refining command output.

The following example first uses the **`file list`** command to display a list of available files, then uses the **`file show`** command to display the contents of the file *config4.cfg*:

```
file list
.ssh
ac-23-ap396-cstest
ac-23-baseline.cfg
ac-23-config
ac-23-config-new
config4.cfg
ipfactory.txt
logfiles
scw_Rel_1.5.101

file show config4.cfg
Time {
    NTPServer1        10.1.11.200;
    NTPServer2        1.1.1.2;
    LocalTimeZoneName PDT;
    Enable            true;
}
Layer3Forwarding {
    Forwarding 1 {
        Enable               true;
        DestIPAddress        0.0.0.0;
        DestSubnetMask       0.0.0.0;
        LANDevice            1;
        IPInterface          1;
        ForwardingGroupIndex 0;
    }
```

```
    ForwardingGroup 1 {
        Name SCW-CLI-Test-String1;
    }
}
[output truncated]
```

5.1.3 Using the *file match* Command

Issue the **`file match <string> <filename>`** command (where <string> can include wildcards) from the Operational Mode to match a text string in the contents of a specified file. The output displays the contents of any full line that contains that string. The match string is case sensitive.

The following example displays all lines with the string *LAN* in the file *sn_configu.cfg*:

```
file match LAN sn_config.cfg
        LANDevice            1;
LANDevice 1 {
    LANHostConfigManagement {
    LANEthernetInterfaceConfig 1 {
LANDevice 2 {
    LANHostConfigManagement {
    LANEthernetInterfaceConfig 1 {
LANDevice 3 {
    LANHostConfigManagement {
    LANEthernetInterfaceConfig 1 {
            PrimaryLANDevice                        1;
            PrimaryLANDevice                        1;
            PrimaryLANDevice                        3;
            PrimaryLANDevice                        1;
```

The following example matches the string *1048576392* in the *debug.log* file in the *logfiles* directory to view the events related to session ID *1048576392*. This command is useful for diagnosing problems related to a given session.

```
file match 1048576392 logfiles/debug.log
  1a 00000001  S 1025 uem      11/09/07 17:09:26.219399 umts_ue_construct: UE:1048576392
  1a 00000001  m 1025 uem      11/09/07 17:09:26.219539 umts_ue_t::handle_rrc_msg:
UE:1048576392 MsgId:CV_ASN_RRC_CONN_REQ Category:4
  1a 00000001  m 1025 uem      11/09/07 17:09:26.219562 umts_ue_t::handle_rrcconnreq:
UE:1048576392 TMSI:1355 CELL:14 State:4
  1a 00000001  M 1025 uem      11/09/07 17:09:26.219568 umts_ue_t::handle_rrcconnreq.
UE:1048576392 Rel Ver:3 EstabCause:origSubscribedTrafficCall.
  1a 00000001  m 1025 uem      11/09/07 17:09:26.219577 umts_ue_t::change_conn_state:
UE:1048576392 Old State:IDLE New State:REQ_RCVD
  1a 00000001  m 1025 uem      11/09/07 17:09:26.219586 umts_ue_t::set_ps_session_type
UE:1048576392 Type:0->3
  1a 00000001  m 1025 uem     11/09/07 17:09:26.219591 umts_ue_t::set_master_session_type
UE:1048576392 Type:0->5
```

The following example matches the string *1048576392* in all files in the *logfiles* directory to view the events related to session ID *1048576392*:

```
file match 1048576392 logfiles/*
```

Refer to Section 2.4.1, Processing Command Output on page 11 and Section 2.4.2, Filtering Output with Regular Expressions on page 12 for information about refining command output.

5.1.4 Using the *file get* Command

Issue the **file get <source-url> <local_path>** command from the Operational Mode to copy a file from a remote server to the services node. This is typically used to download a software package for an upgrade using Secure Copy Protocol (SCP), File Transfer Protocol (FTP), or Trivial File Transfer Protocol (TFTP).

If you do not specify a password for the remote server, a prompt for the password displays. The password you enter is not echoed to the terminal.

The following example uses SCP to copy the software image from the remote server at IP address *10.1.11.30*:

```
file get scp://user@10.1.11.30/volume/prebuild/packages/rel_1.6.1.12/
scw_Rel_1.6.1.12 scw_Rel_1.6.1.12
user@10.1.11.30's password:
```

The following example uses SCP to copy the software image from the remote server at IP address *10.1.11.30* on port *49152*:

```
file get scp://user@10.1.11.30:49152/volume/prebuild/packages/rel_3.1.5.12/
scw_Rel_3.1.5.12 scw_Rel_3.1.5.12
user@10.1.11.30's password:
```

5.1.5 Using the *file put* Command

Issue the **file put <local_path> <destination-url>** command from the Operational Mode to copy a file to a remote server using SCP, FTP, or TFTP. If you do not specify a password for the remote server, a prompt for the password displays. The password you enter is not echoed to the terminal.

The following example uses SCP to copy the file *system.debug.log* to a server:

```
file put logfiles/debug.log scp://user@10.1.11.22/a/system.debug.log
user@10.1.11.22's password:
```

The following example uses SCP to copy the file system.debug.log to a server on port *49152*:

```
file put logfiles/debug.log scp://user@10.1.11.22:49152/a/system.debug.log
user@10.1.11.22's password:
```

The command prompts you to enter your password on the remote server.

5.1.6 Using the *file archive* Command

Issue the **file archive <archive_name>** command from the Operational Mode to archive a file, list of files, or directory in the services node. You can use wildcards to include multiple files.

By default, the archived file is compressed. Use the *NoCompress* parameter to archive without compression. In either case the archived file must be given a filename with a *.tgz* or *.tar.gz* extension.

The following example first uses the **file list** command to display a list of available files, then uses the **file archive** command to archive the files *hs-insert.txt* and *hs-test.txt* into an archive named *newarchive.tar.gz*, and finally uses the **file list** command again to verify the file has been archived:

```
file list
bc_add_cell_test_config
error_incidents/
filecopy-2011-08-26_195437
filecopy-2011-08-26_195437.tgz
filecopy-2011-08-26_201737
hs-insert.txt
```

```
hs-test.txt

file archive newarchive.tar.gz hs*.txt

file list
bc_add_cell_test_config
error_incidents/
filecopy-2011-08-26_195437
filecopy-2011-08-26_195437.tgz
filecopy-2011-08-26_201737
hs-insert.txt
hs-test.txt
newarchive.tar.gz
```

5.1.7 Using the *file delete* Command

Issue the **`file delete <file_to_delete>`** command in the Operational Mode to delete the specified file on the services node.

The following example first uses the **`file list`** command to display a list of available files, then uses the **`file delete`** to delete the file *filecopy-2011-08-26_184214*, and finally uses the **`file list`** command again to verify the file has been deleted:

```
file list
bc_add_cell_test_config
error_incidents/
filecopy-2011-08-26_184214
filecopy-2011-08-26_195437
filecopy-2011-08-26_201737
filecopy-2011-08-26_203909
filecopy-2011-08-26_205837
filecopy-2011-08-26_211307

file delete filecopy-2011-08-26_184214

file list
bc_add_cell_test_config
error_incidents/
filecopy-2011-08-26_195437
filecopy-2011-08-26_201737
filecopy-2011-08-26_203909
filecopy-2011-08-26_205837
filecopy-2011-08-26_211307
```

5.1.8 Using the *file storage cleanup* Command

Issue the **`file storage cleanup`** command from the Operational Mode to delete temporary files from the services node to increase storage space and improve system performance. A list of temporary files displays with a confirmation prompt. Enter **`y`** to delete the files. A file confirmation message displays.

```
file storage cleanup
==================================
The following files will be deleted
==================================

 Size       Name
---------- ------------------------------------------------------------
    798513 /scw/data/error_data/incidents/SN-1025.110824.10_57_38.tgz
    545057 /scw/data/error_data/incidents/SN-1025.110824.11_02_09.tgz
   1109963 /scw/data/error_data/incidents/SN-1025.snmpd.1883.110823.21_58_14.tgz
    244941 /scw/data/logfiles/db_logfile-2011-08-26_070840.log
      6507 /scw/data/logfiles/db_logfile-2011-08-26_101410.log
      6507 /scw/data/logfiles/db_logfile-2011-08-26_145351.log
      8843 /scw/data/logfiles/db_logfile-2011-08-26_184441.log
```

```
       465 /scw/data/logfiles/db_logfile-2011-08-26_184840.log
~~~~~~~~~
~~~~~~~~

Total size = 30067645

Delete these files? [y/N]
y
All files deleted
```

5.1.9 Rotating Debug Log Files

The services node supports one current and up to ten previous debug log files. Each log file is identified by a timestamp. When the current log file reaches 50 megabytes, it is automatically closed (rotated) and a new log begins. If there are already ten old log files, the system deletes the oldest log file.

You can manually rotate a log file by entering the **`request log rotate subsystem debug`** command from the Operational Mode. Optionally, to add a comment to the new log file, first issue the **`request log rotate marker`** command:

`request log rotate subsystem debug marker <textstring>`

5.1.10 Configuring a Remote Server for Log Files

Configure the parameters of up to three remote servers to receive error bundle, debug, and performance management files for archival purposes. To avoid congestion in environments with multiple services nodes you can schedule files to upload at random times.

If the path to the remote directory is not specified, the file transferred to the user's home directory. Refer to the *SpiderCloud OS (SCOS) Administrator Guide* for information about configuring remote servers when using forwarding groups.

Note

You can set the number of upload attempts with the *MaxAttemptDuration* parameter for the number of seconds, or the *MaxAttempts* parameter for the number of tries, but not both.

To configure a remote server for log files

Step 1 From the Configuration Mode, issue the **`set System FileManagement`** command to configure and enable a remote server. In this example:

- it will attempt to send the file for *60* seconds
- it will randomly attempt to send the files after a delay of *4* minutes
- the remote device has the index number *1*
- the upload server has the IP address *10.20.10.1*
- the user name and password are *admin*
- it stores the file in the */a/logfile* directory
- the file transfer uses the *SCP* protocol
- it will attempt to upload the file for one minute (*60* seconds).

`set System FileManagement 1 Enable true ModuleID DebugLog MaxAttemptDuration 60 RandomUploadMaxDelay 4 UploadTarget 1 Host 10.20.10.1 Enable true Username admin Password admin Protocol SCP RemotePath /a/logfiles`

Step 2 Issue the **`show System FileManagement`** command to display configuration information about the remote file server:

```
show System FileManagement 1
Enable               true;
ModuleID             DebugLog;
```

```
MaxAttemptDuration   60;
MaxAttempts          0;
RandomUploadMaxDelay 4;
UploadTarget 1 {
    Enable               true;
    Priority             Primary;
    Protocol             SCP;
    Username             admin;
    Password             admin;
    Host                 10.20.10.1;
    ForwardingGroupIndex 0;
    RemotePath           /a/logfiles;
    RemotePermissions    rw-r--r--;
    OverwriteMode        Overwrite;
}
```

5.1.11 Viewing the Audit Log

The services node maintains a unified audit log of every read, right, and operational commands made to the system. It captures the user that issued the command, the interface used to make the command (CLI, LCI, or SpiderNet), and the date and time the command was issued.

The services node maintains up to 10 audit log files. The active log file is named *scw_audit.log*. Older files are named *scw_audit.log.1* through *scw_audit.log.9*. Each log file contains a maximum of 10 Mb of data. When the current audit log file reaches that maximum size, it is rolled over and renamed *scw_audit.log1*. Older files are renamed to the next higher number and the oldest file is erased.

To view the audit log

Step 1 From the Operational Mode, issue the following command to view the current audit log:

```
file show logfiles/scw_audit.log
```

5.1.12 UMTS Call Performance Event Report Files

The services node can create a file containing call report information with call details which provide per-UE information associated with signaling procedures on the E-RAN. It provides procedure information for operations and customer service teams to track user call performance.

These files can be uploaded to a designated host and can be imported to third-party applications for near real-time call performance monitoring. When properly configured, they can provide detailed per-UE information over user-defined granularity periods. Call performance report files are disabled by default, but can be enabled upon demand.

Each call constitutes multiple rows of the call performance report file. The services node retains call report files to a maximum of 1 GB on the SCSN-9000 and 300 Mb on the SCSN-8000.

The file details:

- the user IMSI
- radio technology
- signaling procedures and results, such as call origination, handovers, and reason for context release
- start/stop timestamps
- associated systems procedures, such as source/target, cells, and MMEs
- KPIs such as peak data rate associated with the procedure.

A call report file is created either at the end of the configured periodic interval or when the size exceeds a configurable size. By default the periodic interval is one minute but can be set to any one minute interval up to 30.

The filename identifies the services node that generated the file, the RAN technology (UMTS or LTE), a sequential serial number, and local timestamp in the following format: cdr.log.umts.<sequenceNumber>.<eNodeBID>.<hostname>.<date>.<time>.gz. For example: *cdr.log.umts.1.5.ac-355.140904.18_45_15*.gz.

All UMTS events are part of the default *UMTSCallPerformanceEventReport* group. Individual events are also members of one more specialized sub-group to target specific network events, such as mobility, enabling CPER filtering and reporting with reduced file size.

The ability to manage logging of events at a group level simplifies CPER log management. Rather than having to enable or disable each CPER event individually, you can enable or disable a specific group of events that are or are not of interest and send the filtered log files to an upload target.

The following UMTS event sub-groups are defined in the system:

- UMTSCallPerformanceEventReportCallManagement: UMTS call management CPER events
- UMTSCallPerformanceEventReportMobility: UMTS UE mobility CPER events (future)
- UMTSCallPerformanceEventReportRANAPMessage: UMTS RANAP message CPER events (future)

UMTSCallPerformanceEventReportRRCMessage: UMTS RRC message CPER events (future)

Refer to the *SpiderCloud Call Performance Event Reporting Guide* for information about CRER events, event attributes and the format of CPER files.

Configuring UMTS call performance event report logs

Step 1 From the Configuration Mode, issue the following command to enable CPER file logging on the UMTS subsystem. This example sets the rollover interval to *1* minute and the file rollover size to *1* Mb.

```
set System CallPerformanceEventReport UMTS Enable true RollOverInte rval 1
RollOverSize 1
```

Step 2 Issue the following command to verify the configuration:

```
show System CallPerformanceEventReport
UMTS {
    Enable           true;
    RollOverInterval 1;
    RollOverSize     1;
}
```

Step 3 From the Configuration Mode, issue the **`set System FileManagement`** command to configure and enable a remote server. In this example:

- set the module ID to *UMTSCallPerformanceEventReport*
- it will attempt to send the file for *60* seconds
- it will randomly attempt to send the files after a delay of *4* minutes
- the remote device has the index number *1*
- the upload server has the IP address *10.20.10.1*
- the user name and password are *admin*
- uses forwarding group *2*
- it stores the file in the */a/logfile* directory
- the file transfer uses the *SCP* protocol

```
set System FileManagement 1 Enable true ModuleID UMTSCallPerformanceEventReport
MaxAttemptDuration 60 RandomUploadMaxDelay 4 UploadTarget 1 Host 10.20.10.1
Enable true Username admin Password admin Protocol SCP RemotePath /a/logfiles
```

Step 4 Issue the **show System FileManagement** command to display configuration information about the remote file server:

```
show System FileManagement 1
Enable               true;
ModuleID             UMTSCallPerformanceEventReport;
MaxAttemptDuration   60;
MaxAttempts          0;
RandomUploadMaxDelay 4;
UploadTarget 1 {
    Enable               true;
    Priority             Primary;
    Protocol             SCP;
    Username             admin;
    Password             admin;
    Host                 10.20.10.1;
}
```

5.1.13 LTE Call Performance Event Report Logs

The services node can create a file containing call report information with call details which provide per-UE information associated with signaling procedures on the E-RAN. It provides procedure information for operations and customer service teams to track user call performance.

These files can be uploaded to a designated host and can be imported to third-party applications for near real-time call performance monitoring. When properly configured, they can provide detailed per-UE information over user-defined granularity periods. Call performance report files are disabled by default, but can be enabled upon demand.

Each call constitutes multiple rows of the call performance report file. The services node retains call report files to a maximum of 1 GB on the SCSN-9000 and 300 Mb on the SCSN-8000.

The file details:

- the user IMSI
- radio technology
- signaling procedures and results, such as call origination, handovers, and reason for context release
- start/stop timestamps
- associated systems procedures, such as source/target, cells, and MMEs
- KPIs such as peak data rate associated with the procedure.

A call report file is created either at the end of the configured periodic interval or when the size exceeds a configurable size. By default the periodic interval is one minute but can be set to any one minute interval up to 30.

The filename identifies the services node that generated the file, the RAN technology (UMTS or LTE), a sequential serial number, and local timestamp in the following format: cdr.log.lts.<sequenceNumber>.<eNodeBID>.<hostname>.<date>.<time>.gz. For example: *cdr.log.lte.1.5.ac-355.140904.18_45_15*.gz.

Refer to the *SpiderCloud Call Performance Event Reporting Guide* for information about CRER events, event attributes and the format of CPER files.

Configuring LTE call performance event report logs

Step 1 From the Configuration Mode, issue the following command to enable CPER file logging on the LTE subsystem. This example sets the rollover interval to *1* minute and the file rollover size to *1* Mb.

```
set System CallPerformanceEventReport LTE Enable true RollOverInte rval 1
RollOverSize 1
```

Step 2 Issue the following command to verify the configuration:

```
show System CallPerformanceEventReport
LTE {
    Enable           true;
    RollOverInterval 1;
    RollOverSize     1;
}
```

Step 3 From the Configuration Mode, issue the **`set System FileManagement`** command to configure and enable a remote server. In this example:

- set the module ID to *LTECallPerformanceEventReport*
- it will attempt to send the file for *60* seconds
- it will randomly attempt to send the files after a delay of *4* minutes
- the remote device has the index number *1*
- the upload server has the IP address *10.20.10.2*
- the user name and password are *admin*
- uses forwarding group *2*
- it stores the file in the */a/logfile* directory
- the file transfer uses the *SCP* protocol

```
set System FileManagement 1 Enable true ModuleID LTECallPerformanceEventReport
MaxAttemptDuration 60 RandomUploadMaxDelay 4 UploadTarget 1 Host 10.20.10.1
Enable true Username admin Password admin Protocol SCP RemotePath /a/logfiles
```

Step 4 Issue the **`show System FileManagement`** command to display configuration information about the remote file server:

```
show System FileManagement 1
Enable               true;
ModuleID             LTECallPerformanceEventReport;
MaxAttemptDuration   60;
MaxAttempts          0;
RandomUploadMaxDelay 4;
UploadTarget 1 {
    Enable               true;
    Priority             Primary;
    Protocol             SCP;
    Username             admin;
    Password             admin;
    Host                 10.20.10.2;
    ForwardingGroupIndex 2;
    RemotePath           /a/logfiles;
}
```

5.1.13.1 LTE Call Performance Event Report Event Filtering

All LTE events are part of the default *LTECallPerformanceEventReport* group. Individual events are also members of one more specialized sub-group to target specific network events, such as mobility, enabling CPER filtering and reporting with reduced file size.

The ability to manage logging of events at a group level simplifies CPER log management. Rather than having to enable or disable each CPER event individually, you can enable or disable a specific group of events that are or are not of interest and send the filtered log files to an upload target.

The following LTE event sub-groups are defined in the system:

- LTECallPerformanceEventReport: the default group containing all LTE CPER events
- LTECallPerformanceEventReportCallManagement: call management CPER events
- LTECallPerformanceEventReportMobility: LTE UE mobility CPER events
- LTECallPerformanceEventReportRRCMessage: LTE RRC message CPER events
- LTECallPerformanceEventReportS1APMessage: LTE S1-AP message CPER events

To filter LTE call performance event report events

Step 1 From the Configuration Mode, issue the following command to filter CPER events by groups. This example creates filter *1* and filters for event groups *LTECallPerformanceEventReportCallManagement* and *LTECallPerformanceEventReportInternalMobility*

```
set System EventManagement Filter 1 Group [
LTECallPerformanceEventReportCallManagement
LTECallPerformanceEventReportInternalMobility ]
```

Step 2 Issue the following command to verify the configuration:

```
show System EventManagement Filter
Filter 1 {
    Group    "[ LTECallPerformanceEventReportCallManagement
LTECallPerformanceEventReportInternalMobility ]";
}
```

Step 3 Issue the following command to associate the filter with an upload target to send the logs to a remote server. This example uses Target *1*.

```
set System EventManagement Target 1 CallPerformanceEventReport CPERModule
LTECallPerformanceEventReport Filter [ 1 ]
```

Step 4 Issue the following command to verify the configuration:

```
show System EventManagement Target 1 CallPerformanceEventReport CPERModule
CPERModule LTECallPerformanceEventReport;
```

5.2 Operational Mode Show Commands

Operational Mode show commands are discussed in detail in Chapter 6, “Show Commands” on page 65.

5.3 The Fetch Command

The Operational Mode **`fetch`** command is an efficient method of retrieving first-level system configuration and status information. It follows the structure of the data model and returns the immediate attributes of the specified object path.

The fetch command syntax consists of a list of objects optionally followed by an attribute name and a list of object indexes. The fetch command allows for one omitted index for a list object in an object path. The omitted index must be the last one in the object path.

Data model objects are separated by a period (**.**) For example:

QueueManagement.ControlPlaneCoS.

Using the fetch command in the CLI, replace the period with a space:

```
fetch QueueManagement ControlPlaneCoS
```

```
ControlPlaneClassificationGroup  0;
DefaultSignalingClassQueue       7;
DefaultOAMClassQueue             3;
```

The fetch command will retrieve attributes from multiple instances of an object domain. The following example returns the *EthernetId* of all provisioned radio nodes:

```
fetch RadioNode EthernetID
RadioNode 10 {
        EthernetID                      11:22:33:44:55:66;
}
RadioNode 384 {
        EthernetID                      00:24:48:01:2a:28;
}
```

5.3.1 Indexes

The lower-case *i* enclosed in curly braces (**{i}**) is an index that is a variable integer representing an array structure. For example:

RadioNode.{i}.

In the CLI, you can omit the index number and search for all of the specified objects and their first-level attributes:

```
fetch RadioNode
RadioNode 10 {
        Enable                          true;
        EthernetID                      11:22:33:44:55:66;
        SecurityMode                    secure;
        OperState                       OOS-NOTPRESENT;
        LANDeviceNumberOfEntries        1;
        ForwardingEngineNumberOfEntries 1;
}
RadioNode 384 {
        Enable                          true;
        EthernetID                      00:24:48:01:2a:28;
        SecurityMode                    open;
        OperState                       OOS-NOTPRESENT;
        LANDeviceNumberOfEntries        1;
        ForwardingEngineNumberOfEntries 1;
}
```

Each indexed object has an index attribute with the prefix of that object and a suffix of *Index*. For example: **RadioNode RadioNodeIndex.** Specify the unique object index number for a more targeted search. The following example searches for first-level information about radio node *10*:

```
fetch RadioNode RadioNodeIndex 10
Enable                          true;
EthernetID                      11:22:33:44:55:66;
SecurityMode                    secure;
OperState                       OOS-NOTPRESENT;
LANDeviceNumberOfEntries        1;
ForwardingEngineNumberOfEntries 1;
```

Refer to *SCOS Data Model Reference Guide* for complete details about objects and parameters that comprise the system configuration and operational state. Refer to the *SpiderCloud OS (SCOS) CLI User Guide* for information about how to map the data model to the CLI hierarchy.

5.4 Request Commands

Execute request commands, also known as imperative commands, to make system-wide requests. All request commands originate from the Operational Mode. The CLI contains the following request commands:

- **Interface:** Interface management commands
- **Statistics:** Statistics management operations
- **airlink:** Air link operations
- **clear-debug:** Clear debug filters
- **diagnostics:** Diagnostic mode commands
- **forwarding:** Forwarding operations
- **ipsec:** IPsec operations
- **lci:** LCI installation report
- **log:** Log file operations
- **lte:** LTE operations
- **management-server:** Services node commands to SpiderNet
- **message:** Send message to terminal of one or all users
- **port-mirroring:** Port mirroring operations
- **radionode:** Radio node operations
- **scheduled:** Scheduled actions
- **services-module:** Service module operations
- **set-debug:** Set debug filters
- **system:** System operations
- **test:** Test commands
- **umts:** Cellular operations
- **wlan:** W-Fi operations (currently not supported)

Request commands are hierarchical in structure. Table 10 shows each request command, its sub-commands, and description:

Table 10: Request Commands

Request Command	Sub-Command	Sub-Command	Description
Interface	Statistics	reset	Interface management commands
Statistics	cell	refresh reset	Statistics operations related to cell
	delete	all	Delete all the stored statistics
	lte	reset	Resets LTE statistics
	refresh	all	Updates all the stored statistics
	reset	all	Reset all the statistics collected until now
	serviceavailability	reset	Resets parameters related to service availability KPIs
	session	reset rollsnapshot	Statistics operations related to resetting the session statistics
	syslog	reset	Reset syslog statistics
	system	refresh reset	Statistics operations related to the system
	ue	reset	Statistics operations related to this UE
	wlan	reset	Statistics operations related to WLAN services. Not currently supported.
airlink	log	mask phymon-filter	Set an air link logging mask or physical layer monitoring filter
clear-debug	***many***		Clear debug filtering options
log	bundle	count subsystem	Create an error-bundle
	mark	marker subsystem	Mark the log with a marker message
	rotate	marker maxfiles subsystem	Rotate the log with a marker message
	tail		Continually view additions to the end of log file

Table 10: Request Commands *(continued)*

Request Command	Sub-Command	Sub-Command	Description
lte			
	cell	disable enable	LTE cell operations
	core	reset	Resets the LTE core connection
	debug	m3ap - M3AP Layer debug options s1ap - S1AP Layer debug options sctp - Stream Control Transmission Protocol layer debug options tucl - TCP/UDP Connection Layer debug options x2ap - X2AP Layer debug options	LTE debug settings
	rem	start stop	Start or stop a REM scan operation
	self-config	anr-neighbors detected-neighbors neighborlist-create stale-neighbors tx-power-assignment	LTE REM scanning operations
	ue	session	Clear the LTE UE session
	ue-info-req	disable enable	Enable or disable sending LTE RRC UE Information Request message
	x2	reset	Resets the LTE X2 connections
management-server	connection-abort	timeout	Abort an existing connection to the management server
	disable		Disable the management server
	inform		Initiate inform to management server
	logs	delete disable enable	Enable, disable, or delete management server logs
message	admin		Send a message to the administrator
	all		Send a message to all logged in users
port-mirroring	disable		Disable port mirroring
	enable		Enable port mirroring

Table 10: Request Commands *(continued)*

Request Command	Sub-Command	Sub-Command	Description
radionode	led	follow locate normal	Radio node LED related control commands to follow an IMSI, locate a specific radio node, and reset LED display properties
	replace	node_id with_node_id	Replace one radio node with another
	swap	node_id node_mac with_node_id with_node_mac	Commands for swapping a radio node with another radio node.
set-debug	***many***		Set debug filters
scheduled	actions	start status stop	Schedule or stop a radio node link test or REM scan or monitor the status of a scheduled task
services-module	PendingService	register remove	Register or remove a pending service
	service-image	get	Retrieve the service image from a remote host

Table 10: Request Commands *(continued)*

Request Command	Sub-Command	Sub-Command	Description
system	Database	Backup Clone Restore	Backup, clone and schedule for file upload, or restore the system database
	banner	load	Set system banner shown at login
	certificate	CACert LocalCert	Configure the certificate type
	diagnostics		For SpiderCloud use only
	file	transfer	File management commands
	import	RN-Cell-config	Import a configuration file to configure radio nodes and cells
	logout	user	Log a current user out of the session
	ping	Host	Execute a ping test
	reboot	clean (services node) node (radio node) node-wlan (currently disabled)	Reboot the system with the options to clear the services node database or the database of an individual radio node with the options listed below this table*
	revert		Revert the software image
	ssh	hostkey	Generates a new ssh host key
	update	LCI-dir clean package	Updates the system software with the options listed below this table*
	usb	format mount unmount	USB drive actions. Not currently supported.

Table 10: Request Commands *(continued)*

Request Command	Sub-Command	Sub-Command	Description
test	ip	rn-link	Start, stop, or clear results of a radio node benchmarking test
	lte	add-cell anr_exe_cell anr_neighbor_pair detectedextcell detectedneighbor self-config simul_anr_message	LTE internal cell, external neighbor cell and REM scan actions
	mem-dump		Start an internal memory dump
	nmc-dump		Start an NMC internal state dump
	snmp	trap	Sends SNMP test traps
	umts	add-cell detectedextcell detectedneighbor	UMTS internal cell and external neighbor cell actions
umts	cell	disable enable maxtxpower stats	UMTS cell operations
	core	reset	Resets the UMTS core
	debug	***many***	UMTS debug settings
	rem	start stop	Start or stop a REM scan operation
	self-config	clear-ue-measurements neighborlist-create start tx-power-assignment	Operations related to UMTS REM scanning
	ue	bler-measurement sessions stats	UMTS UE commands
	zone	detectedlist referenceneighborhood	UMTS zone operations
wlan	web	web	Displays WLAN information about the local host. Currently not supported.

* The following describes the system actions for each system reboot and update option:

- **clean-configuration:** Replaces the existing configuration with the factory configuration
- **clean-data:** Removes all persistent data such as the configuration, log files, core files
- **clean-db:** Removes the *cellmgr-ini* file
- **clean-none:** Removes nothing, the system will retain the existing configuration

5.5 Additional Utilities

The section discusses Operational Mode system utilities:

5.5.1 id

Use the **id** command to view user ID information

```
id
user = admin(9000), gid=900, groups=admin-group, gids=900
```

5.5.2 ping

Use the **ping <ip_address>** command to send an ICMP ECHO_REQUEST to a network host:

```
ping 10.1.11.200
PING 10.1.11.200 (10.1.11.200) 56(84) bytes of data.
64 bytes from 10.1.11.200: icmp_seq=1 ttl=63 time=2.45 ms
64 bytes from 10.1.11.200: icmp_seq=2 ttl=63 time=0.167 ms
64 bytes from 10.1.11.200: icmp_seq=3 ttl=63 time=0.151 ms
```

5.5.3 set

Run the **set** command to configure the following CLI properties:

- set autowizard: Enable/disable automatic query for mandatory elements

```
set autowizard true
```

- complete-on-space: Enable/disable completion on space

```
set complete-on-space true
```

- idle-timeout: Configure idle timeout in seconds

```
set idle-timeout 180
```

- paginate: Paginate output from CLI commands

```
set paginate false
```

- show: Show default values when showing the configuration

```
set show defaults true
```

- system: Set system properties

```
set system clock 2012-04-12T13:30:00
```

5.5.4 source

Use the **source <file>** command to instruct the system to execute the commands in the defined source file. This essentially executes a script.

```
source cli_commands.txt
```

5.5.5 test policy IMSI

Use the **test policy IMSI <IMSI>** command to view the details of the Policy of the defined IMSI:

```
test policy IMSI 001010123451204
Policy result for a master session for IMSI 001010123451204
    Policy group index: 1
    Policy trace: [ 1 ]
```

```
Action: Reject
Policy switching mode: PassThrough
FM Switching mode: PassThrough
Passthrough Uplink ClassificationGroup index: 0, nexthop: 0
Passthrough Downlink ClassificationGroup index: 0, nexthop: 0
Local-switching Uplink ClassificationGroup index: 0, nexthop: 0
Local-switching Downlink ClassificationGroup index: 0, nexthop: 0
Switching filter nexthop: 23
Primary LANDevice: 1
Primary IPInterface: 1
Forwarding group: 0
Default class queue: -1
Admission priority group: 0
```

6 Show Commands

This chapter contains the following sections:

6.1 Show Command Overview

Show commands are a valuable method for surveilling the state of the system and troubleshooting problems. They provide read-only access to the SpiderCloud system operational state that includes:

- state variables and alarms
- data that change slowly and may be cached
- statistics and counters
- data that change frequently, are not cached, and are retrieved upon demand

There are separate sets of show commands for each command mode:

- The Configuration Mode show commands follow the outline of the data model. Refer to Chapter 3, “Structure of the Data Model.” on page 21 for information about the data model format and its structure. Refer to Section 4.2, Command Mode Show Commands on page 37 for information about Configuration Mode show commands.
- The Operational Mode contains two types of show commands:
  - The regular Operational show commands are specially developed to provide concise output in tabular format.
  - The special **`show status OpState`** commands follow the outline of the data model but provide a more detailed look at the system configuration than the Configuration Mode show commands.

6.2 Custom Show Commands

The following custom Operational Mode **show** commands were developed to provide information useful in monitoring and managing the SpiderCloud system. Refer to the *SCOS NB Data Model Reference Guide* for information about these commands and their parameters.

6.2.1 Show Command Output Truncation

Many of the show command outputs have been shortened in this chapter for brevity and ease of reading. You can also filter command output to view specific information that might otherwise be buried in massive command output. Refer to Section 2.4.1, Processing Command Output on page 11 for more information about filtering show command output.

6.2.2 Brief, Detailed, and Verbose Command Versions

Where noted, some commands have brief, detailed, and verbose parameters that filter or expand the output of a show command. For demonstrative purposes, example output for a sampling of these commands are mixed into this section.

6.2.3 show Cell

Use the **`show Cell`** command to view information for provisioned cells:

```
show Cell
CellHandle  Name              RN     Radio  ModeInUse    ConfState    OperState
----------  ----------------  -----  -----  -----------  -----------  ---------
         1  near-bathroom         1      1  UMTSNodeB    PROVISIONED  IS-IDLE
         2  near-Printer-24       2      1  UMTSNodeB    PROVISIONED  IS-IDLE
         3  PaulM-office          3      1  UMTSNodeB    PROVISIONED  IS-IDLE
         4  near-cube-56          4      1  UMTSNodeB    PROVISIONED  IS-IDLE
         5  near-Bills-Off        5      1  UMTSNodeB    PROVISIONED  IS-IDLE
         6  near-Paresh-cube      6      1  UMTSNodeB    PROVISIONED  IS-IDLE
         7  near-103              7      1  UMTSNodeB    PROVISIONED  IS-IDLE
         8  sn-8k-lab             8      1  UMTSNodeB    PROVISIONED  IS-IDLE
```

6.2.4 show Cell CellHandle

Use the **`show Cell CellHandle <CellHandle>`** command to view detailed information about a defined cell:

```
show Cell CellHandle 2
CellHandle  Name              RN     Radio  ModeInUse    ConfState    OperState
----------  ----------------  -----  -----  -----------  -----------  ------------------
         2  near-Printer-24       2      1  UMTSNodeB    PROVISIONED  IS-IDLE
```

6.2.5 show Cell UMTS

Use the **`show Cell UMTS`** command to view UMTS information for cells. This command also has **`Detail`** and **`Verbose`** versions.

```
show Cell UMTS
CellHandle  Name              RN     CID    CellID      PSC  MaxTxPwr  ModeInUse     RLs
----------  ----------------  -----  -----  ----------  ---  --------  ------------  ---
         1  near-bathroom         1      1    65601537  100    4.0dBm  UMTSNodeB       0
         2  near-Printer-24       2      2    65601538  101    0.0dBm  UMTSNodeB       0
         3  PaulM-office          3      3    65601539  102    1.0dBm  UMTSNodeB       2
         4  near-cube-56          4      4    65601540  103    2.0dBm  UMTSNodeB       1
         5  near-Bills-Off        5      5    65601541  104    2.0dBm  UMTSNodeB       0
         6  near-Paresh-cube      6      6    65601542  105   -4.0dBm  UMTSNodeB       9
         7  near-103              7      7    65601543  106    5.0dBm  UMTSNodeB       0
         8  sn-8k-lab             8      8    65601544  107   -2.0dBm  UMTSNodeB       3
```

6.2.6 show cli

Use the **`show cli`** command to view CLI settings:

```
show cli
autowizard true;
complete-on-space true;
display-level 99999999;
history 100;
idle-timeout 0;
output {
    file terminal;
}
paginate true;
```

```
screen {
    length 46;
    width 80;
}
show {
    defaults false;
}
terminal xterm;
```

6.2.7 show cli history

Use the **`show cli history`** command to view the CLI command history:

```
show cli history
09:38:14 -- set idle-timeout 0
09:39:42 -- show Cell
09:40:18 -- show Cell CellHandle
09:40:42 -- show Cell CellHandle
09:40:55 -- show Cell CellHandle UMTS
09:41:53 -- show Cell UMTS CellHandle
09:42:04 -- show Cell UMTS CellHandle
09:42:21 -- show cli
09:42:30 -- show cli history
```

6.2.8 show configuration

Use the **`show configuration`** command to view details of the current configuration. Note that entering this command without a parameter returns all system information. Enter one of the following parameters to filter the information for a more useful return:

- Cell: A table containing the configured system cell list
- DeviceInfo: General device information
- FAPService: Femto Access Point (FAP) service object
- LANDevice: Port number as labeled on device faceplate
- Layer3Forwarding: Forwarding configuration
- Layer3Routing: Routing configuration
- ManagementDevice: Management device port number
- ManagementServer: Parameters related to the services node ACS connection
- PacketCapture: Packet capture settings
- QueueManagement: Queuing and classifications (ACLs)
- RadioNode: Parameters relating to all provisioned radio nodes
- ServicesHosts: Services host parameters
- ServicesNode: Parameters relating to the services node
- System: Parameters relating to the entire system
- Time: System time and NTP related parameters
- WLANService: Parameters relating to WLAN services
- details: Show details of the system configuration
- displaylevel: Level depth to show output

6.2.8.1 Examples

`show configuration Layer3Forwarding`

```
Forwarding 1 {
    Enable               true;
    DestIPAddress        0.0.0.0;
    DestSubnetMask       0.0.0.0;
    LANDevice            1;
    IPInterface          1;
}
Forwarding 2 {
    Enable               true;
    DestIPAddress        172.17.0.0;
    DestSubnetMask       255.255.255.0;
    GatewayIPAddress     172.16.0.2;
    LANDevice            2;
    IPInterface          1;
}
```

```
show configuration PacketCapture
Enable false;
Mode   All;
MaximumFileSize 16;
Filter {
    PeerAPort         40005;
    Protocol          17;
}
```

```
show configuration Time
NTPServer1        10.1.11.200;
NTPServer2        0.0.0.0;
Enable            true;
```

6.2.9 show Core

Use the **show Core** command to view the connectivity status between the services node and the provider core network:

```
show Core

IPSec:
    SecGWServer 1: Local 172.17.0.8 <-> Remote 172.17.0.1 (Established)
        SN Internal IPAddress: 172.19.0.24

Control:
    Protocol: Iuh
    Iuh gateway (Connected):
        Peering: Local 172.19.0.24:29169 <-> Remote 192.168.10.3:29169
```

6.2.10 show Core Control

Use the **show Core Control** command to view the core control connection status:

```
show Core Control
Protocol: Pico
CSDomain (Connected):
    Peering: Local 10.1.80.200:1706 <-> Remote 10.1.80.9:1705
PSDomain (Connected):
    Peering: Local 10.1.80.200:4097 <-> Remote 10.1.80.5:1702
```

6.2.11 show Core IPSec

Use the **show Core IPSec** command to view the core IPsec connection status:

```
show Core IPSec
SecGWServer 1: 10.1.193.2 <-> 10.1.214.20 (Established)
```

6.2.12 show Core IPSec Detail

Use the **show Core IPSec Detail** command to view detailed IPsec core information:

```
show Core IPSec Detail
SecGWServer: 1 (10.1.214.20)
    Status: Established
    LastTriedTime: 2012-03-23T16:53:56Z
    LocalIPAddress: 10.1.193.2
    ChildSA: 56 (outbound)
        SPI: 54799320, CreationTime: 2012-03-23T16:53:56Z, NextRekeyTime: 2012-03-23T17:05:56Z, ExpirationTime:
2012-03-23T17:08:56Z
        ESPEncryptInUse: AES-CBC, ESPIntegrityInUse: HMAC-SHA1-96
        TrafficBytes: 672, TrafficPackets: 8
        IntegrityErrors: 0, ReplayErrors: 0, CryptErrors: 0, DecryptErrors: 0, SAErrors: 0, PolicyErrors: 0,
SoftLifeErrors: 0
    ChildSA: 57 (inbound)
       SPI: 3448491707, CreationTime: 2012-03-23T16:53:56Z, NextRekeyTime: 2012-03-23T17:06:41Z, ExpirationTime:
2012-03-23T17:08:56Z
        ESPEncryptInUse: AES-CBC, ESPIntegrityInUse: HMAC-SHA1-96
        TrafficBytes: 672, TrafficPackets: 8
           IntegrityErrors: 0, ReplayErrors: 0, CryptErrors: 0, DecryptErrors: 0, SAErrors: 0, PolicyErrors: 0,
SoftLifeErrors: 0
```

6.2.13 show Core IPSec Pkey

Use the **show Core IPSec Pkey** command to view the services node Pkey certificate:

```
show Core IPSec Pkey
Pkey 1: my_cert.pem
    Description: SN's Certificate
    SerialNumber: 82E2
    Subject: C=US, ST= , L= , O=SpiderCloud Wireless, OU= ,
CN=002448FFFF0000f0.servicesnode.spidercloud.com/
emailAddress=certifying_authority@spidercloud.com
    subjectAlt: 002448FFFF0000f0.servicesnode.spidercloud.com
    Issuer: C=US, ST= , L= , O=SpiderCloud Wireless, OU= , CN=SCW_Issuer_CA/
emailAddress=certifying_authority@spidercloud.com
    NotBefore: 2011-12-20T05:51:51Z, NotAfter: 2016-12-18T05:51:51Z
    PrivateKeyExists: true
```

6.2.14 show debug

Use the **show debug** command to view debug related information:

```
show debug
module        group   level   node
-----------------------------------
FC            PF_KEY  status  0
FE            ALL     minor   0
IPC           ALL     minor   0
CD            ALL     major   0
NB            ALL     major   0
PM            ALL     major   0
AD            ALL     major   0
AF            ALL     major   0
NEM           ALL     major   0
```

6.2.15 show FAPService 1 FAPControl UMTS HomeNodeB

Use the **show FAPService 1 FAPControl UMTS HomeNodeB** command to view information about the Home NodeB registration:

```
show FAPService 1 FAPControl UMTS HomeNodeB
RegistrationTimeout   1;
UERegistrationTimeout 1;
UEIdleTimeout         3600;
```

6.2.16 show Forwarding Interface

Use the **show Forwarding Interface** command to view forwarding interface information:

```
show Forwarding Interface
Interface 4:
    Name: ge-1, LANDevice: 1, InternalInterfaceID: 1, Flags: 7, FIBID: 254,
    MTU: 1500, VLANTag: 0, MACAddress: 00:24:48:00:50:e0,
    NextHopStack: [ 11 10 ]
Interface 5:
    Name: ge-2, LANDevice: 2, InternalInterfaceID: 2, Flags: 7, FIBID: 254,
    MTU: 1500, VLANTag: 0, MACAddress: 00:24:48:00:50:e0,
    NextHopStack: [ 11 10 ]
Interface 6:
    Name: ge-3, LANDevice: 3, InternalInterfaceID: 3, Flags: 2, FIBID: 254,
    MTU: 1500, VLANTag: 0, MACAddress: 00:24:48:00:50:e0,
    NextHopStack: [ 11 10 ]
```

6.2.17 show Forwarding NextHop

Use the **show Forwarding NextHop** command to view the forwarding next hop information. This command also has **Brief**, **Detail**, and **Verbose** versions.

```
show Forwarding NextHop
NextHop  NexthopType
-------  -----------
      1  Local
      2  Drop
      3  DNS
      4  Filter
      5  Filter
      6  Filter
      7  UDPTunnel
      8  Filter
      9  Filter
```

6.2.18 show Forwarding NextHop Detail

Use the **show Forwarding NextHop Detail** command to see detailed next hop information:

```
show Forwarding NextHop Detail
NextHop 1:
    NexthopType: Local

NextHop 2:
    NexthopType: Drop

NextHop 3:
    NexthopType: DNS

NextHop 4:
    NexthopType: Filter
    Name: GenericDNSmatch, Role: Unspecified, FIBID: 0
        Rule: Priority: 0, Type: Complex,
            IPProtocol: 17
```

```
            SourceIPAddress: 0.0.0.0, SourceIPMask: 0.0.0.0
            SourcePort: 0, SourcePortEnd: 0
            DestIPAddress: 0.0.0.0, DestIPMask: 0.0.0.0
            DestPort: 53, DestPortEnd: 53
            Action: COUNTER counter=0, NHForwardingOp: Replace,
            NextHopStack: [ 3 ]
```

6.2.19 show Interface

Use the **`show Interface`** command to view information about IP and Ethernet interfaces. This command also has **`Brief`**, **`Detail`**, and **`Verbose`** versions.

```
show Interface
LANDevice 1:
    Enable: true, MACAddress: 00:24:48:00:50:e0, OperState: IS-NORMAL,
    Status: Up
    IPInterface 1:
        Enable: true, IPInterfaceIPAddress: 10.1.80.200,
        IPInterfaceSubnetMask: 255.255.255.0, VLANID: 0,
        ForwardingGroupIndex: 0, OperState: IS-NORMAL
LANDevice 2:
    Enable: true, MACAddress: 00:24:48:00:50:e0, OperState: IS-NORMAL,
    Status: Up
    IPInterface 1:
        Enable: true, IPInterfaceIPAddress: 172.17.0.2,
        IPInterfaceSubnetMask: 255.255.255.0, VLANID: 0,
        ForwardingGroupIndex: 0, OperState: IS-NORMAL
```

6.2.20 show Interface IPInterface

Use the **`show Interface IPInterface <InterfaceNumber>`** command to view information about the IP interface of a specified services node Ethernet port. This command also has **`Brief`**, **`Detail`**, and **`Verbose`** versions.

```
show Interface IPInterface 1
LANDevice 1:
    Enable: true, MACAddress: 00:24:48:00:50:e0, OperState: IS-NORMAL,
    Status: Up
    IPInterface 1:
        Enable: true, IPInterfaceIPAddress: 10.1.80.200,
        IPInterfaceSubnetMask: 255.255.255.0, VLANID: 0,
        ForwardingGroupIndex: 0, OperState: IS-NORMAL
LANDevice 2:
    Enable: true, MACAddress: 00:24:48:00:50:e0, OperState: IS-NORMAL,
    Status: Up
    IPInterface 1:
        Enable: true, IPInterfaceIPAddress: 172.17.0.2,
        IPInterfaceSubnetMask: 255.255.255.0, VLANID: 0,
        ForwardingGroupIndex: 0, OperState: IS-NORMAL
```

6.2.21 show Interface IPInterface 1 Verbose

Use the **`show Interface IPInterface 1 Verbose`** command to view the maximum amount of detail about the IP interface of the specified services node Ethernet port:

```
show Interface IPInterface 1 Verbose
LANDevice 1:
    Enable: true, MACAddress: 00:24:48:00:50:e0, OperState: IS-NORMAL,
    Status: Up
    BytesSent:            2582874709  BytesReceived:             626465615
    PacketsSent:            10278598  PacketsReceived:             9612516
    ErrorsSent:                    0  ErrorsReceived:                    0
    UnicastPacketsSent:     10278572  UnicastPacketsReceived:      9608181
    DiscardPacketsSent:            0  DiscardPacketsReceived:            0
    MulticastPacketsSent:          0  MulticastPacketsReceived:       4209
    BroadcastPacketsSent:         26  BroadcastPacketsReceived:        126
```

```
    IPInterface 1:
        Enable: true, IPInterfaceIPAddress: 10.1.80.200,
        IPInterfaceSubnetMask: 255.255.255.0, VLANID: 0,
        ForwardingGroupIndex: 0, OperState: IS-NORMAL, DHCPServerEnable: false
        BytesSent:            2582221452  BytesReceived:             626465738
        PacketsSent:            10265519  PacketsReceived:             9612518
        ErrorsSent:                    0  ErrorsReceived:                    0
        UnicastPacketsSent:     10265519  UnicastPacketsReceived:      9608183
        DiscardPacketsSent:            0  DiscardPacketsReceived:            0
        MulticastPacketsSent:          0  MulticastPacketsReceived:       4209
        BroadcastPacketsSent:          0  BroadcastPacketsReceived:        126
```

6.2.22 show Interface LANDevice

Use the **`show Interface LANDevice <InterfaceNumber>`** command to view the interfaces for the specified Ethernet port.

```
show Interface LANDevice 5
LANDevice 5:
    Enable: true, MACAddress: 00:24:48:00:50:e0, OperState: OOS-FAULT,
    Status: NoLink
    IPInterface 1:
        Enable: true, IPInterfaceIPAddress: 5.5.5.5,
        IPInterfaceSubnetMask: 255.255.255.0, VLANID: 0,
        ForwardingGroupIndex: 0, OperState: OOS-INHERITED
```

6.2.23 show IP ARP

Use the **`show IP ARP`** command to view information about the Address Resolution Protocol (ARP) table:

```
show IP ARP
Address                 HWtype  HWaddress           Flags Mask            Iface
172.17.0.198            ether   00:24:48:00:00:2d   C                     ge-2
172.17.0.167            ether   00:24:48:01:2a:3f   C                     ge-2
172.17.0.162            ether   00:24:48:01:2a:26   C                     ge-2
10.1.15.2               ether   00:22:bd:94:45:53   C                     ge-1
10.1.15.5               ether   00:22:bd:94:45:53   C                     ge-1
10.1.11.26              ether   00:22:bd:94:45:53   C                     ge-1
172.17.0.168            ether   00:24:48:01:2a:0a   C                     ge-2
172.17.0.175            ether   00:24:48:01:2a:1f   C                     ge-2
172.17.0.164            ether   00:24:48:01:2a:46   C                     ge-2
172.17.0.169            ether   00:24:48:01:2a:1c   C                     ge-2
172.17.0.161            ether   00:24:48:00:00:47   C                     ge-2
```

6.2.24 show IP Route

Use the show IP Route command to view basic information about the configured static routes:

```
show IP Route
RIB 1, 4 destinations
    10.0.0.0/8
        *[Connect/1] 2013-02-05T17:02:33Z
            > via LANDevice 1, IPInterface 1
    127.0.0.0/16
        *[Connect/1] 2013-02-05T17:02:17Z
            >
    127.1.0.0/16
        *[Connect/1] 2013-02-05T17:02:17Z
            >
    172.30.30.0/24
        *[Connect/1] 2013-02-05T17:02:37Z
            > via LANDevice 2, IPInterface 1
```

6.2.25 show IP Route Configured

Use the **show IP Route Configured** command to view information about the configured static routes to the default gateway:

```
show IP Route Configured
DestIPAddress    GatewayIPAddress  DestSubnetMask   LANDevice  IPInterface  Enable
---------------  ----------------  ---------------  ---------  -----------  ------
0.0.0.0          0.0.0.0           0.0.0.0          1          1            true
172.17.0.0       172.16.0.2        255.255.255.0    2          1            true
```

6.2.26 show IP Route Configured Detail

Use the **show IP Route Configured Detail** command to view detailed information about the static route to the default gateway:

```
show IP Route Configured Detail
Forwarding 1:
    DestIPAddress: 172.17.0.0, DestSubnetMask: 255.255.255.0
    GatewayIPAddress: 172.16.0.2, LANDevice: 2, IPInterface: 1,
    ForwardingGroupIndex: 0

Forwarding 2:
    DestIPAddress: 10.3.19.0, DestSubnetMask: 255.255.255.0
    GatewayIPAddress: 0.0.0.0, ManagementDevice: 0, LANDevice: 8,
    IPInterface: 1, ForwardingGroupIndex: 81
```

6.2.27 show RadioNode

Use the **show RadioNode** command to display the number, name, MAC address, IP addresses, and operational state of each radio node in the system:

```
show RadioNode
RN    Name              Enable  EthernetID         IPAddress        OuterIPAddress   OperState
----  --------------  ------  -----------------  ---------------  ---------------  -----------
   1  ap-820            true    00:24:48:01:2a:eb  172.17.0.107     172.17.0.107     IS-NORMAL
   2  ap-830            true    00:24:48:01:2a:e1  172.17.0.108     172.17.0.108     IS-NORMAL
   4  ap-816            true    00:24:48:01:2a:f2  172.17.0.106     172.17.0.106     IS-NORMAL
```

6.2.28 show RadioNode Radio

Use the **show RadioNode Radio** command to display information about the parameters of all radios in the system:

```
show RadioNode Radio
RN    Radio  Enable  Band           ActualBand     OperState
----  -----  ------  -------------  -------------  ------------------
   1      1  true    umts-band-IV   umts-band-IV   IS-NORMAL
   2      1  true    umts-band-IV   umts-band-IV   IS-NORMAL
   3      1  true    umts-band-IV   umts-band-IV   IS-NORMAL
   4      1  true    umts-band-IV   not-available  OOS-NOTPRESENT
   5      1  true    umts-band-IV   not-available  OOS-NOTPRESENT
   7      1  true    umts-band-IV   umts-band-IV   IS-NORMAL
   8      1  true    umts-band-IV   umts-band-IV   IS-NORMAL
   9      1  true    umts-band-IV   umts-band-IV   IS-NORMAL
  10      1  true    umts-band-IV   umts-band-IV   IS-NORMAL
  11      1  true    umts-band-IV   umts-band-IV   IS-NORMAL
```

6.2.29 show RFMgmt UMTS

Use the **show RFMgmt UMTS** command with its parameters to display aspects of the current RF management state. Top level parameters for this command are:

- **Configuration:** Show configuration
- **DetectedCells:** Show detected cells
- **MeasurementOfGSMCell:** Show internal cells that detected GSM cells
- **MeasurementOfUMTSCell:** Show internal cells that detected UMTS cells
- **NeighborCells:** Show neighbor cells

```
show RFMgmt UMTS MeasurementOfGSMCell
Detecting CellID  Detecting CID  BSIC  CI     ARFCN  BandID    RSSI

---------------  ------------  ----  -----  -----  --------  ----

       131072001             1    25   5176    128   GSM 850   -80

       131072001             1    27  32004    129   GSM 850   -69

       131072004             4    25   5176    128   GSM 850   -75

       131072004             4    27  32004    129   GSM 850   -68
```

```
show RFMgmt UMTS Configuration
    Global Configuration:
    .....................

    FAPService REM Configuration:
    =============================================
    WCDMAFDD:
      Scan Periodically                       false
      Periodic Interval                       86400
      Periodic Time                           1970-01-04T00:00:00Z
      DL UARFCN List                          -
      Periodic TxPwr Refresh                  false
      Periodic TxPwr Refresh Interval         86400
    GSM:
      Periodic Interval                       86400
      REM Band List                           -
      ARFCN List                              -

    FAPService REM Scans:
    =====================================
    GSM                REM Scan          true
    UMTS Ext IntraFreq REM Scan          true
    UMTS Ext InterFreq REM Scan          true
    UMTS Int IntraFreq REM Scan          true

    FAPService Locks:
    ================================
    RF               Lock          false
    Neighbor List    Lock          false
    PSC              Lock          false
    Max FAP Tx Power Lock          false
[output truncated]
```

```
show RFMgmt UMTS DetectedCells

List Of Cells Detected By Internal Cell With Cell Handle 1, CID 1, And Cell ID 65536001:
--------------------------------------------------------------------------------

Detected INTERNAL UMTS Cells:
============================
```

```
  CID  Cell Handle     Cell ID  PSC  DL UARFCN  CPICH RSCP*
-----  -----------  ----------  ---  ---------  -----------
    2            2    65536002    2       1962          -74
    3            3    65536003    3       1962          -93

* Measured When Detected Internal UMTS Cell Was Transmitting At FAPService
Maximum MaxFAPTxPower

Detected EXTERNAL UMTS Cells:
=============================
  CID  Cell Handle     Cell ID  PSC  DL UARFCN  PCPICHTxPower  CPICH RSCP
-----  -----------  ----------  ---  ---------  -------------  ----------

Detected GSM Cells:
===================
Cell Handle  ARFCN  Frequency Band  BSIC     CI  RSSI
-----------  -----  --------------  ----  -----  ----


List Of Cells Detected By Internal Cell With Cell Handle 2, CID 2, And Cell ID 65536002:
--------------------------------------------------------------------------------
Detected INTERNAL UMTS Cells:
=============================
  CID  Cell Handle     Cell ID  PSC  DL UARFCN  CPICH RSCP*
-----  -----------  ----------  ---  ---------  -----------
    1            1    65536001    1       1962          -79
    3            3    65536003    3       1962          -84

* Measured When Detected Internal UMTS Cell Was Transmitting At FAPService
Maximum MaxFAPTxPower

Detected EXTERNAL UMTS Cells:
=============================
  CID  Cell Handle     Cell ID  PSC  DL UARFCN  PCPICHTxPower  CPICH RSCP
-----  -----------  ----------  ---  ---------  -------------  ----------

Detected GSM Cells:
===================
Cell Handle  ARFCN  Frequency Band  BSIC     CI  RSSI
-----------  -----  --------------  ----  -----  ----

[output truncated]
```

6.2.30 show Route

Use the **show Route** command to view routing table information.

```
show Route
Destination  Gateway  SubnetMask  LANDevice  IPInterface  Enable
-------------------------------------------------------------------
0.0.0.0      0.0.0.0  0.0.0.0     1          1            true
```

6.2.31 show ServicesNode

Use the **show ServicesNode** command to view the state of the services node:

```
show ServicesNode
SN    ArriveTime            OperState
----  --------------------  ------------------
1025  2011-09-27T23:57:35Z  IS-NORMAL
```

6.2.32 show ServicesNode Resource

Use the **show ServicesNode Resource** command to view information about the current resource usage of the services node:

```
show ServicesNode Resource
ServicesNode 1025:
    CPU:
        User:           7.14%
        Kernel:         2.88%
        IOWait:         0.00%
        Swap:           0.00%
        Idle:          89.96%
        LoadAvg1:       0.03
        LoadAvg5:       0.08
        LoadAvg15:      0.07
    Memory:
        Free:          87.17%
        Used:          12.83%
        Cache:         54.06%
        Total:       1690048K
```

6.2.33 show ServicesNode Time

Use the **show ServicesNode Time** command to view the current services node time, time the services node came online, and time in service.

```
show ServicesNode Time
ServicesNode 1025:
    CurrentTime: 2012-03-23T11:53:30Z
    ArriveTime:  2012-03-22T22:19:53Z
    UpTime:      13:33:37
```

6.2.34 show Session

Use the **show Session** command to display all active UE sessions. This command also has **Brief**, **Detail**, and **Verbose** versions.

```
show Session
Session   IMSI             D  V  ConnectTime        Type
--------  ---------------  -  -  -----------------  ----
  926107  001010123451204  0  1  01-11 18:39:39.98  UMTS
  926106  001010123451065  1  0  01-11 18:39:39.64  UMTS
  926104  001010123451342  0  1  01-11 18:39:32.30  UMTS
  926101  001010123451285  0  1  01-11 18:39:29.11  UMTS
  926098  001010123451134  0  0  01-11 18:39:20.79  UMTS
  925973  001010123451133  1  0  01-11 18:35:50.88  UMTS
  925897  001010123451388  0  1  01-11 18:33:26.53  UMTS
  925894  001010123451385  0  1  01-11 18:33:22.05  UMTS
```

6.2.35 show Session Detail UEIPAddress

Use the **show Session Detail UEIPAddress** command to display information about all voice and data connections for the defined IP address of a connected device. This command is useful for locating the IMSI of the device, which can then be used for access control and in debugging procedures.

```
show Session Detail UEIPAddress 10.1.80.174
Session: 173384
    RATType: UMTS, IMSI: 001010123451269
    NumberOfActiveDataSessions: 1, NumberOfActiveVoiceSessions: 0
    DataSessionNumberOfEntries: 1, VoiceSessionNumberOfEntries: 0
    ConnectTime: 2011-09-21T14:52:38.550501Z
    RRCState: Cell_PCH, UMTSSessionID: 1048579917, CurrentSnapshotID: 24, ServingCellHandle: 14
    CSDomainActive: false, CSSessionType: UNKNOWN
    PSDomainActive: true, PSSessionType: HSUPA
    DataSession: 173385
```

```
        FlowID: 483, APNName: 11apn1, SwitchingMode: PassThrough
        UEIPAddress: 10.1.80.174
        PrimaryDNSIPAddress: 0.0.0.0, SecondaryDNSIPAddress: 0.0.0.0, ProviderPrimaryDNSIPAddress: 10.1.11.200,
        ProviderSecondaryDNSIPAddress: 10.1.11.200
    RLC DTCH stats:
        RadioBearer: 5
            RLCMode: AM, RBType: DTCH, IsActive: true
    Handover Statistics:
        NumServingCellChanges: 3, NumAsetAdds: 7, NumAsetDeletes: 2, NumAsetSwaps: 0
    RadioLinks:
        RadioLink: 1
            CellHandle: 14, PSC: 206, RLID: 0, IsActive: false, IsServingCell: 1
```

6.2.36 show Session Detail UENATIPAddress

Use the **`show Session Detail UENATIPAddress`** command to display information about all voice and data connections for each IMSI in a given NATted IP address of a connected device. This command is useful for locating the IMSI of the device, which can then be used for access control and in debugging procedures.

`show Session Detail UENATIPAddress 172.20.0.108`

```
Session: 173384
    RATType: UMTS, IMSI: 001010123451355
    NumberOfActiveDataSessions: 1, NumberOfActiveVoiceSessions: 0
    DataSessionNumberOfEntries: 1, VoiceSessionNumberOfEntries: 0
    ConnectTime: 2011-09-21T14:55:18.550501Z
    RRCState: Cell_DCH, UMTSSessionID: 1048579935, CurrentSnapshotID: 12, ServingCellHandle: 4
    CSDomainActive: false, CSSessionType: UNKNOWN
    PSDomainActive: true, PSSessionType: HSUPA
    DataSession: 173385
        FlowID: 483, APNName: apn3, SwitchingMode: NAPT
        UEIPAddress: 10.1.80.155, UENATIPAddress: 172.20.0.108
        PrimaryDNSIPAddress: 0.0.0.0, SecondaryDNSIPAddress: 0.0.0.0, ProviderPrimaryDNSIPAddress: 10.1.11.200,
        ProviderSecondaryDNSIPAddress: 10.1.11.200
    RLC DTCH stats:
        RadioBearer: 5
            RLCMode: AM, RBType: DTCH, IsActive: true
    Handover Statistics:
        NumServingCellChanges: 3, NumAsetAdds: 7, NumAsetDeletes: 2, NumAsetSwaps: 0
    RadioLinks:
        RadioLink: 1
            CellHandle: 4, PSC: 201, RLID: 0, IsActive: false, IsServingCell: 1
```

6.2.37 show Session History

Use the **`show Session History`** command to view current and historical UE session information. This command also has **`Brief`**, **`Detail`**, and **`Verbose`** versions.

`show Session History`

```
Session   IMSI             D  V  ConnectTime        DisconnectTime     Type
--------  ---------------  -  -  -----------------  -----------------  ----
  299516  001010123451358  0  0  03-23 11:55:26.29  03-23 11:55:27.97  UMTS
  299515  0010101234568 55  0  0  03-23 11:53:30.88  03-23 11:53:34.73  UMTS
  299514  001010123451331  0  0  03-23 11:52:38.38  03-23 11:52:39.78  UMTS
  299513  001010123451231  0  0  03-23 11:52:02.02  03-23 11:52:04.04  UMTS
  299512  001010123451264  0  0  03-23 11:51:52.43  03-23 11:51:53.85  UMTS
  299511  001010123451330  0  0  03-23 11:51:42.08  03-23 11:51:43.47  UMTS
  299510  001010123451231  0  0  03-23 11:47:46.03  03-23 11:47:50.10  UMTS
  299506  001010123451373  0  0  03-23 11:39:22.93  03-23 11:50:16.63  UMTS
```

6.2.38 show Session IMSI

Use the **`show Session IMSI <IMSI>`** command to view information about a defined IMSI. This command also has **`Brief`**, **`Detail`**, and **`Verbose`** versions.

`show Session IMSI 001010123451374`

```
Session   IMSI             D  V  ConnectTime        Type
--------  ---------------  -  -  -----------------  ----
  299496  001010123451374  1  1  03-23 11:36:25.42  UMTS
```

6.2.39 show Session IMSI Detail

Use the **show Session IMSI <IMSI> Detail** command to view detailed information about a defined IMSI:

```
show Session IMSI 001010123451374 Detail
Session: 299496
    RATType: UMTS, IMSI: 001010123451374
    NumberOfActiveDataSessions: 1, NumberOfActiveVoiceSessions: 1
    DataSessionNumberOfEntries: 1, VoiceSessionNumberOfEntries: 1
    ConnectTime: 2012-03-23T11:36:25.424839Z
    RRCState: Cell_DCH, UMTSSessionID: 1049625757, CurrentSnapshotID: 1392,
    ServingCellHandle: 8
    CSDomainActive: true, CSSessionType: Voice_FR
    PSDomainActive: true, PSSessionType: HSPA
    DataSession: 299502
        FlowID: 320, APNName: internet, SwitchingMode: PassThrough
        UEIPAddress: 10.1.80.197
        PrimaryDNSIPAddress: 0.0.0.0, SecondaryDNSIPAddress: 0.0.0.0,
        ProviderPrimaryDNSIPAddress: 10.1.11.200,
        ProviderSecondaryDNSIPAddress: 10.1.11.200
    VoiceSession: 299497
        FlowID: 317
    RLC DTCH stats:
        RadioBearer: 8
            RLCMode: AM, RBType: DTCH, IsActive: true
    Handover Statistics:
        NumServingCellChanges: 1387, NumAsetAdds: 1387, NumAsetDeletes: 1386,
        NumAsetSwaps: 1029
    RadioLinks:
        RadioLink: 1
            CellHandle: 8, PSC: 107, RLID: 1, IsActive: true, IsServingCell: 1
```

6.2.40 show Session IMSI Verbose

Use the **show Session IMSI <IMSI> Verbose** command to view all details of a defined IMSI:

```
show Session IMSI 001010123451370 Verbose
Session: 306061
    RATType: UMTS, IMSI: 001010123451370
    NumberOfActiveDataSessions: 0, NumberOfActiveVoiceSessions: 1
    DataSessionNumberOfEntries: 0, VoiceSessionNumberOfEntries: 1
    ConnectTime: 2012-03-26T15:51:47.924248Z, ConnectCause: Voice,
    DisconnectCause: Still Active
    UMTSSessionID: 1049624620, CurrentSnapshotID: 37, ServingCellHandle: 3,
    AdmissionControlPriority: 65533
    CSDomainActive: true, CSSessionType: Voice_FR
    PSDomainActive: false, PSSessionType: UNKNOWN
    VoiceSession: 306062
        FlowID: 19
    RLC DTCH stats:
        RadioBearer:
        ----------------------------
        Downlink:
            SDUsIngress:
            SDUsDropped:
```

6.2.41 show Session IMSI History

Use the **show Session IMSI <IMSI> History** command to view the history of the defined IMSI. This command also has **Brief**, **Detail**, and **Verbose** versions.

```
show Session IMSI 001010123451374 History
Session   IMSI             D  V  ConnectTime        DisconnectTime     Type
--------  ---------------  -  -  -----------------  -----------------  ----
  299496  001010123451374  1  1  03-23 11:36:25.42  -                  UMTS
  299489  001010123451374  0  0  03-23 11:36:02.54  03-23 11:36:17.60  UMTS
```

```
  299476  001010123451374  0  0  03-23 11:33:49.42  03-23 11:35:28.29  UMTS
  299439  001010123451374  0  0  03-23 11:21:18.57  03-23 11:22:25.73  UMTS
  299395  001010123451374  0  0  03-23 10:55:41.48  03-23 11:06:24.79  UMTS
```

6.2.42 show Session UMTS

Use the **show Session UMTS** command to view UTMS session information for each currently active IMSI. This command also has **Brief**, **Detail**, and **Verbose** versions.

show Session UMTS

```
Session   IMSI            D  V  ConnectTime        RRCState    ConnectCause    Cell   CSSessionType  PSSessionType
--------  --------------- -  -  -----------------  ----------  --------------  -----  -------------  -------------
  299500  -                0  1  03-23 11:36:53.25  Cell_DCH    Voice              6  Voice_FR       UNKNOWN
  299498  001010123451370  0  1  03-23 11:36:48.70  Cell_DCH    Voice              3  Voice_FR       UNKNOWN
  299496  001010123451374  1  1  03-23 11:36:25.42  Cell_DCH    Voice              8  Voice_FR       HSPA
  299494  001010123451205  1  1  03-23 11:36:20.61  Cell_DCH    Voice              6  Voice_FR       HSPA
  299488  001010123451258  0  1  03-23 11:36:02.17  Cell_DCH    Voice              6  Voice_FR       UNKNOWN
  299486  001010123451357  1  1  03-23 11:35:58.14  Cell_DCH    Voice              6  Voice_FR       HSDPA_64U
```

6.2.43 show Session UMTS Verbose

Use the **show Session UMTS Verbose** command to view all details of each currently active IMSI:

show Session UMTS Verbose

```
Session: 306089
    RATType: UMTS, IMSI: 001010123451329
    NumberOfActiveDataSessions: 0, NumberOfActiveVoiceSessions: 1
    DataSessionNumberOfEntries: 0, VoiceSessionNumberOfEntries: 1
    ConnectTime: 2012-03-26T15:59:39.037546Z, ConnectCause: Voice,
    DisconnectCause: Still Active
    RRCState: Cell_DCH, UMTSSessionID: 1049624634, CurrentSnapshotID: 13,
    ServingCellHandle: 7, AdmissionControlPriority: 65533
    CSDomainActive: true, CSSessionType: Voice_FR
    PSDomainActive: false, PSSessionType: UNKNOWN
    VoiceSession: 306090
        FlowID: 33
    RLC DTCH stats:
        RadioBearer:
        ----------------------------
        Downlink:
            SDUsIngress:
            SDUsDropped:
```

6.2.44 show Session UMTS History

Use the **show session UMTS History** command to view current and historical UE session information. This command also has **Brief**, **Detail**, and **Verbose** versions. Additionally, it has the following sub-commands:

- CellHandle: Show only UMTS UE sessions with at least one radio link on a cell matching this CellHandle
- IMSI: Show only UMTS UE sessions matching this IMSI
- SessionID: Show only UE sessions matching this SessionID
- UEIPAddress: Show only UMTS UE sessions with a data session with this IP address
- UENATIPAddress: Show only UMTS UE sessions with a data session with this NATted IP address

show Session UMTS History

```
Session   IMSI            D  V  ConnectTime        RRCState    ConnectCause    Cell   DisconnectTime     DisconnectCause
--------  --------------- -  -  -----------------  ----------  --------------  -----  -----------------  ---------------
  301243  001010123451256  0  0  03-24 11:19:44.46  Cell_DCH    Registration       3  03-24 11:19:46.49  Normal Release
  301242  001010123451359  0  0  03-24 11:19:13.37  Cell_DCH    HSDPA              3  -                  Still Active
  301241  001010123456856  0  0  03-24 11:16:54.88  Cell_DCH    Registration       6  03-24 11:16:56.28  Normal Release
  301240  001010123451356  0  0  03-24 11:16:30.36  Cell_DCH    Registration       4  03-24 11:16:32.01  Normal Release
  301239  001010123451255  0  0  03-24 11:15:56.96  Cell_DCH    Registration       3  03-24 11:15:58.50  Normal Release
  301237  001010123451373  0  0  03-24 11:14:17.99  Cell_FACH   HSUPA              6  03-24 11:14:29.63  Normal Release
  301236  001010123451373  0  0  03-24 11:13:43.27  Cell_DCH    HSUPA              6  03-24 11:14:15.57  Normal Release
```

6.2.45 show Session UMTS Detail SessionID

Use the show **`Session UMTS Detail SessionID`** command to display detailed information about a specific UE session:

```
show Session UMTS Detail SessionID 1832428
SessionID   IMSI          RRCState           ConnectTime          ConnectCause         DisconnectTime       DisconnectCause
----------- --------------- ------------------ -------------------- -------------------- -------------------- ----------
1832428     234159103675269 Cell_DCH           2011-09-28T13:58:21.817106Z HSDPA             -                    Still Active

Status:
    CSDomainActive:          false
    CSSessionType:           UNKNOWN
    PSDomainActive:          true
    PSSessionType:           HSDPA
    AdmissionControlPriority: 65534
    CSGStatus:               Authorized
    UMTSSessionID:           904921657
    CurrentSnapshotID:       1

Session Info:
    DataSessionNumberOfEntries: 1
    VoiceSessionNumberOfEntries: 0
    DataSession: 1832429
        FlowID:                      120
        SwitchingMode:               NAPT
        APNName:                     -
        UEIPAddress:                 10.172.30.110
        UENATIPAddress:              10.15.11.112
        ProviderPrimaryDNSIPAddress:   0.0.0.0
        ProviderSecondaryDNSIPAddress: 0.0.0.0
        PrimaryDNSIPAddress:         8.8.8.8
        SecondaryDNSIPAddress:       0.0.0.0
        ForwardingGroupIndex:        0

RLC DTCH Stats:
    RadioBearerID:                       5
    ---------------------------  ----------
    RLCMode:                            AM
    RBType:                           DTCH
    IsActive:                         true
    Downlink:                            -
        SDUsIngress:                    11
        SDUsDropped:                     0
        DataPDUsSent:                   55
        NumStatusPDUsReceived:          10
```

6.2.46 show Session UMTS Summary

Use the show Session UMTS Summary command to view specific information a session. It can be filtered by cell, user session, IP address, or NATted IP address. You can also view the session history.

```
show Session UMTS Summary
Total sessions (active 8, peak 28)

CS sessions (active 3, peak 15):
    EmergencyCall: active 0, peak 0
    Voice: active 2, peak 14
    VideoTelephony: active 0, peak 0
    SMS: active 0, peak 0
    Registration: active 1, peak 4

PS sessions (active 5, peak 14):
    R99Data: active 1, peak 1
    HSDPA: active 2, peak 2
    HSUPA: active 2, peak 12
    Registration: active 0, peak 4
```

6.2.47 show status

Use the **`show status`** command to view the current system status. Note that entering this command without a parameter returns the status of all objects in the system. This can be extremely detailed. The following parameters filter the output. Most of these have one or more parameters below them for additional filtering.

- OpState: Read only view of configuration and run time state (refer to Using Show Status OpState on page 85)

- Statistics: Statistics management operations
- airlink: Air link operations
- core: Core level commands
- debug: Show debug settings
- displaylevel: Depth to show
- process-details: Show process-details
- processes: Show process list
- system: System operations
- umts: UMTS operations
- wlan: WLAN operations

6.2.47.1 Example

```
show status process-details
process-details 1304 {
    name          node-mgr;
    state         RUNNING;
    file-name     /load/platform/bin/nodemgr;
    service-name  nodemgr;
    restartable   0;
    restart-count 0;
    shell         false;
    grp-name      Platform-nodemgr;
    grp-state     RUNNING;
    scw-status    0;
    wstatus       0;
}
```

6.2.48 show System Alarm

Use the **`show System Alarm`** command to view the current system alarms:

```
show System Alarm
Object               Name                      ID          Severity  Time
-------------------  ------------------------  ----------  --------  --------------------
LANDevice.3          LINK_DOWN                 553667603   Major     2012-03-22T22:20:19Z
LANDevice.4          LINK_DOWN                 553668627   Major     2012-03-22T22:20:19Z
LANDevice.5          LINK_DOWN                 553669651   Major     2012-03-22T22:20:19Z
LANDevice.6          LINK_DOWN                 553670675   Maj
```

6.2.49 show System Alarm History

Use the **`show System Alarm History`** command to view system historical alarm events:

```
show System Alarm History
Object               ID          Severity       Type          Time
-------------------  ----------  -------------  ------------  --------------------
RadioNode.1          134218756   Cleared        ClearedAlarm  2012-03-22T22:23:38.032346Z
RadioNode.7          134224900   Cleared        ClearedAlarm  2012-03-22T22:23:37.453260Z
RadioNode.2          134219780   Cleared        ClearedAlarm  2012-03-22T22:23:30.755313Z
RadioNode.8          134225924   Cleared        ClearedAlarm  2012-03-22T22:23:25.708982Z
RadioNode.3          134220804   Cleared        ClearedAlarm  2012-03-22T22:23:23.285903Z
RadioNode.6          134223876   Cleared        ClearedAlarm  2012-03-22T22:23:18.299796Z
RadioNode.5          134222852   Cleared        ClearedAlarm  2012-03-22T22:23:15.444536Z
RadioNode.4          134221828   Cleared        ClearedAlarm  2012-03-22T22:22:42.807391Z
RadioNode.1          134218761   Cleared        ClearedAlarm  2012-03-22T22:22:18.373172Z
RadioNode.1          134218756   Major          NewAlarm      2012-03-22T22:22:42.315501Z
```

6.2.50 show System Condition

Use the **`show System Condition`** command to view system-wide conditions:

show System Condition

```
Object                Severity  Name                          State    Change Time
--------------------  --------  ----------------------------  -------  --------------------
System                Indeterminate  RFMGMT_UNEXPECTED_CONFIG    Active   2012-03-22T22:20:45Z
ServicesNode.1025     Major     CORE_IPSEC_TERM               Active   2012-03-22T22:19:49Z
LANDevice.3           Major     LINK_DOWN                     Active   2012-03-22T22:20:19Z
LANDevice.4           Major     LINK_DOWN                     Active   2012-03-22T22:20:19Z
LANDevice.5           Major     LINK_DOWN                     Active   2012-03-22T22:20:19Z
```

6.2.51 show System Event

Use the **show System Event** command to view current system-wide events. This command also has an **Ascending** parameter.

show System Event

```
2012-03-24T11:31:19.587144Z M EVENT_ADMIN_AAA_LOGOUT [MOI="ServicesNode.1025" Username="admin"
PID="22182" Description="admin from process 22182 has logged out." ]
2012-03-24T11:31:18.984291Z I EVENT_ADMIN_AAA_MGMT_SESSION_END [MOI="ServicesNode.1025"
Username="admin" PID="22182" Service="cli" MgmtSessID="628" Description="admin from process 22182
has ended a cli session identified by 628." ]
2012-03-24T11:30:14.033510Z I EVENT_ADMIN_AAA_MGMT_SESSION_BEGIN [MOI="ServicesNode.1025"
Username="admin" PID="22182" Service="cli" MgmtSessID="628" Description="admin from process 22182
has begun a cli session identified by 628." ]
2012-03-24T11:30:13.993161Z M EVENT_ADMIN_AAA_LOGIN [MOI="ServicesNode.1025" Username="admin"
PID="22182" Description="admin has logged in via process identified by 22182." ]
2012-03-24T11:30:07.466149Z M EVENT_ADMIN_AAA_LOGOUT [MOI="ServicesNode.1025" Username="admin"
PID="22111" Description="admin from process 22111 has logged out." ]
2012-03-24T11:30:06.852557Z I EVENT_ADMIN_AAA_MGMT_SESSION_END [MOI="ServicesNode.1025"
Username="admin" PID="22111" Service="cli" MgmtSessID="627" Description="admin from process 22111
has ended a cli session identified by 627." ]
2012-03-24T11:29:56.326773Z I EVENT_ADMIN_AAA_MGMT_SESSION_BEGIN [MOI="ServicesNode.1025"
Username="admin" PID="22111" Service="cli" MgmtSessID="627" Description="admin from process 22111
has begun a cli session identified by 627." ]
2012-03-24T11:29:56.279173Z M EVENT_ADMIN_AAA_LOGIN [MOI="ServicesNode.1025" Username="admin"
PID="22111" Description="admin has logged in via process identified by 22111." ]
2012-03-24T11:26:32.958177Z M EVENT_ADMIN_AAA_LOGOUT [MOI="ServicesNode.1025" Username="admin"
PID="21899" Description="admin from process 21899 has logged out." ]
```

6.2.52 show System Event Count

Use the **show System Event Count** command to view a defined number of current system-wide events. This command also has an **Ascending** parameter.

show System Event Count 2

```
2012-03-24T13:05:13.778822Z I EVENT_ADMIN_AAA_MGMT_SESSION_BEGIN [MOI="ServicesNode.1025"
Username="admin" PID="28621" Service="cli" MgmtSessID="653" Description="admin from process 28621
has begun a cli session identified by 653." ]
2012-03-24T13:05:13.739161Z M EVENT_ADMIN_AAA_LOGIN [MOI="ServicesNode.1025" Username="admin"
PID="28621" Description="admin has logged in via process identified by 28621." ]
2012-03-24T13:01:33.412168Z M EVENT_ADMIN_AAA_LOGOUT [MOI="ServicesNode.1025" Username="admin"
PID="28170" Description="admin from process 28170 has logged out." ]
2012-03-24T13:01:32.810051Z I EVENT_ADMIN_AAA_MGMT_SESSION_END [MOI="ServicesNode.1025"
Username="admin" PID="28170" Service="cli" MgmtSessID="652" Description="admin from process 28170
has ended a cli session identified by 652." ]
2012-03-24T13:00:51.785172Z M EVENT_ADMIN_AAA_LOGOUT [MOI="ServicesNode.1025" Username="admin"
PID="28100" Description="admin from process 28100 has logged out." ]
```

6.2.53 show System File Target

Use the **show System File Target** command to display the status of file upload targets:

show System File Target

```
ModuleID         Host             Priority    Enable  MaxAttemptDuration  MaxAttempts  FailedAttempts
---------------  ---------------  ----------  ------  ------------------  -----------  --------------
DebugLog         10.1.11.17       Primary     true    0                   10                        22
```

6.2.54 show System File Transfer History

Use the show System File Transfer History command to view all files transferred since the last reboot:

```
show System File Transfer History
TransID  ModuleID              Status        RequestCompleteTime    FailedAttempts  LastError
-------  --------------------  ------------  ---------------------  --------------  ---
1024     DebugLog              Complete      2011-09-28T13:32:39Z               0  -
1025     DebugLog              Complete      2011-09-28T14:55:19Z               0  -
1026     DebugLog              Complete      2011-09-28T15:11:06Z               0  -
1027     DebugLog              Complete      2011-09-28T15:17:35Z               0  -
1028     DebugLog              Complete      2011-09-28T15:25:07Z               0  -
1029     DebugLog              Complete      2011-09-28T15:29:40Z               0  -
1030     DebugLog              Complete      2011-09-28T16:03:44Z               0  -
1031     DebugLog              Complete      2011-09-28T16:56:00Z               0  -
1032     DebugLog              Complete      2011-09-28T17:13:03Z               0  -
1033     DebugLog              Complete      2011-09-28T18:07:16Z               0  -
1034     DebugLog              Complete      2011-09-28T18:41:14Z               0  -
1035     DebugLog              Complete      2011-09-28T19:09:32Z               0  -
1036     DebugLog              Complete      2011-09-28T20:03:37Z               0  -
1037     DebugLog              Complete      2011-09-28T20:19:14Z               0  -
1038     DebugLog              Complete      2011-09-28T20:53:12Z               0  -
```

6.2.55 show System Syslog

Use the **show System Syslog** command to view syslog target information. This command has the following parameters:

- Alarm: show system alarms
- Certificate:
- Condition: show system-wide conditions
- Event: show events
- File:
- Syslog: show Syslog target information
- UMTS: show UMTS Session information

```
show System Syslog
Target      Enable  IPAddress        TxMsgs      DiscardMsgs  Filter
----------  ------  ---------------  ----------  -----------  ------------------
         1  true    10.1.12.33             1757            0  [ 1 ]
         2  true    10.1.12.33                0         1785  [ 2 ]
         3  true    10.1.11.26             1757            0  [ 1 ]
         4  true    10.1.11.26                0         1785  [ 2 ]
```

6.2.56 show System UMTS

Use the **show System UMTS** command to view UMTS session information. This command also has **Detail** and **Verbose** versions.

```
show System UMTS
Current Status:
    TimeOfLastStatsReset:    2012-03-22T22:20:42.568605Z
    TimeOfLastStatsUpdate:   2012-03-24T13:46:01.711734Z
    NumCellsProvisioned:     8
    NumCellsActive:          4
    IuCSStatus:              Connected
    IuPSStatus:              Connected
    EmergencyCallActive:     false
```

6.2.57 show UE Location

Use the **show UE Location** command to view most recent location information for all active users. This command has a **Detail** parameter.

```
show UE Location
UEID   IMSI             IMEI             LastUpdate         LastSessionID  LastServingCell
-----  ---------------  ---------------  -----------------  -------------  ---------------
    1  001010123451348  -                -                         211592               -
```

```
    2  001010123451177   -               -                       250265              -
    3  001010123451358   -               03-24 11:36:49.56       301269              4
    4  001010123451349   -               03-24 11:36:49.56       211600              4
    5  001010123451228   -               03-24 11:36:49.56       238563              4
    6  001010123451082   -               03-24 11:36:49.56       238130              4
    7  001010123451080   -               03-24 11:36:49.56       238133              4
```

6.2.58 show UE Location IMSI

Use the **`show UE Location IMSI <IMSI>`** command to view location information about the defined UE:

```
show UE Location IMSI 001010123456952
UEID   IMSI             IMEI            LastUpdate         LastSessionID  LastServingCell
-----  ---------------  --------------  -----------------  -------------  ---------------
    9  001010123456952  -
```

6.2.59 show UE Location Detail

Use the **`show UE Location Detail`** command to view detailed information for all active users:

```
show UE Location Detail
UEID: 1
    IMSI:          001010123451348  LastSessionID:        211592
    IMEI:                        -  LastServingCell:           -

UEID: 2
    IMSI:          001010123451177  LastSessionID:        250265
    IMEI:                        -  LastServingCell:           -

UEID: 3
    LastUpdate: 2012-03-27T10:26:55.72947Z
    IMSI:          001010123451358  LastSessionID:        306844
    IMEI:                        -  LastServingCell:           4
    Last Serving RN: 4
```

6.2.60 show users

Use the **`show users`** command to display a list of the currently active CLI session users.

```
show users
SID  USER  CTX FROM       PROTO LOGIN
 233 admin cli 10.1.80.9  ssh   11:35:02
*201 admin cli 10.1.10.89 ssh   09:38:10
```

6.2.61 show Version

Use the **`show Version`** command to view information about the system software version. This command has a **`Detail`** parameter.

```
show Version
Product  Image    Version  Timestamp
-------  -------  -------  ---------------------
SCOS     running  2.0.0    2012-03-26T21:21:16Z
```

6.2.62 show Version Revert

Use the **`show Version Revert`** command to display information about the system software revert version.

```
show Version Revert
PackageID  Version             Builder     BuildTime
---------  ------------------  ----------  ----------------------------
PLAT       4.0.0.              david       Fri Feb 21 08:22:12 2014 PST
```

```
UMTS       4.0.0.              ortiz       Fri Feb 21 08:50:11 2014 PST
```

6.3 Using Show Status OpState

The Operational Mode *OpState* object contains the complete operational state and state history of the system. The system OpState has the same hierarchy as the data model, and is its top-level object. The OpState hierarchy that contains both read-write and read-only parameters:

- is a superset of the configuration hierarchy (it shares the same skeleton structure)
- includes both configuration and read-only parameters (the whole data model)
- is presented as a read-only (a read-only view of the entire data model)

The **`show status OpState`** command provides a complete dump of all operational state in the hierarchy of the data model. Sub-commands of **`show status OpState`** are equivalent to an SNMP MIB walk starting at a specific place in the data model, and are a superset of Configuration Mode **`show`** commands. For example, compare the results of this command:

```
admin> show status OpState FAPService 1 FAPControl UMTS Gateway
FAPGWPort          29169;
CNConnectionEnable true;
FAPLocalIPAddress  0.0.0.0;
FAPLocalPort       1024;
CNProtocol         Pico;
```

to this command:

```
admin@% show FAPService 1 FAPControl UMTS Gateway
FAPGWPort          29169;
CNConnectionEnable true;
FAPLocalPort       1024;
CNProtocol         Pico;
```

The **`show status OpState`** command contains the `FAPLocalIPAddress` object that is not part of the Configuration Mode show command.

The following are the top-level objects of OpState hierarchy:

- **Cell:** Table containing the configured UMTS system cell list
- **DeviceInfo:** General device information
- **Diagnostics:** Parameters related to diagnostics
- **FAPService:** Femto Access Point (FAP) service object
- **LANDevice:** Port number as labeled on device faceplate
- **LANDeviceNumberOfEntries:** Number of configured Ethernet ports in the system
- **Layer3Forwarding:** Forwarding configuration
- **Layer3Routing:** Routing configuration
- **LTECell:** LTE cell configuration
- **ManagementDevice:** Management device port number
- **ManagementDeviceNumberOfEntries:** Number of configured services node management devices
- **ManagementServer:** Parameters relating to the CPE's association with an ACS
- **NumberOfActiveSessions:** Number of active UE session records
- **PacketCapture:** Packet capture
- **QueueManagement:** Queuing and classifications (ACLs)
- **RadioNode:** Parameters relating to radio nodes

- **ServicesHosts:** Services host parameters
- **ServicesNode:** Parameters relating to services nodes
- **Session:** Parameters relating to UE session
- **SessionNumberOfEntries:** Number of UE session records
- **System:** Parameters relating to the entire system
- **Time:** System time and NTP related parameters
- TrunkDevice: The trunk device port number
- **TrunkDeviceNumberOfEntries:** Number of trunk devices in the services node
- **UE:** Parameters relating to the UE
- **UENumberOfEntries:** Number of UE records
- **WLANServices:** Parameters related to WLAN services


Note

The SpiderCloud data model contains over 3000 attributes and the operating state of a system is complex. The output of the **`show status OpState`** command used without modifiers can run hundreds of thousands of lines. Use this command with the only the most specific modifiers.

The three examples below demonstrate the use of increasingly refined **`show status OpState`** commands.

6.3.1 Two-Level Filter

The following example shows the truncated output of a two-level **`show status OpState`** command filter. It filters first on *FAPService*, then upon *UMTS* (this release supports only FAPService 1, so the 1 does not filter any output). The output was 1057 lines.

```
show status OpState FAPService 1 UMTS
RANAPNumberOfEntries 2;
CurrentStatus {
    NumCellsConfigured           8;
    NumCellsProvisioned          8;
    NumCellsActive               6;
    NumCellsAtAdmissionCtrlLimit 0;
    EmergencyCallActive          false;
    NumActiveSessions            18;
    NumActiveMRAB                6;
    NumActiveCellFACH            0;
    NumActiveCellPCH             1;
    ActiveCSSessions {
        Sum 16;
        Type {
            EmergencyCall  0;
            Voice          16;
            VideoTelephony 0;
            SMS            0;
            Registration   0;

[Output Truncated]
```

6.3.2 Three-Level Filter

The following example shows the truncated output of a three-level **`show status OpState`** command filter. It filters first on *FAPService*, then upon *UMTS*, then upon *SessionManagement*. The output was 112 lines.

```
show status OpState FAPService 1 UMTS SessionManagement
MasterSessionCreation {
    Sum      29637;
    MaxRate  15;
    RateHist "[ 23023 1 0 0 0 0 0 0 0 0 ]";
    Cause {
        EmergencyCall  4;
        Voice          454;
        VideoTelephony 0;
        SMS            0;
        R99Data        0;
        HSDPA          9949;
        HSUPA          3989;
        HardHandIn     0;
        Registration   15241;
    }
}
MasterSessionDeletion {
    NormalRelease 28893;
    Sum           29537;
    AbnormalRelease {
        Sum 631;
        Cause {

[Output Truncated]
```

6.3.3 Four-Level Filter

The following example shows the output of a four-level **`show status OpState`** command filter. It filters first on *FAPService*, then upon *UMTS*, then upon *SessionManagement*, and finally upon *NASSessionDeletion*. The output was 7 lines.

```
show status OpState FAPService 1 UMTS SessionManagement NASSessionDeletion
Sum 36018;
CS {
    Sum 7619;
}
PS {
    Sum 28399;
}
```