



Cisco Fabric Analysis and Correlation Toolkit User Guide

Release 1.0

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Customer Order Number:
Text Part Number: OL-15645-01

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

CCDE, CCENT, Cisco Eos, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, the Cisco logo, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn is a service mark; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0805R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

Cisco Fabric Analysis and Correlation Toolkit User Guide
© 2008 Cisco Systems, Inc. All rights reserved.



CONTENTS

Preface v

Audience v

Organization v

Related Documentation vi

Conventions vi

Obtaining Documentation and Submitting a Service Request vii

CHAPTER 1

Understanding the Fabric Analysis and Correlation Toolkit 1-1

About FACT 1-1

How FACT Works 1-2

Scanning and Reporting 1-2

Repository 1-2

Connections 1-3

Network Maintenance 1-3

Hardware and Software Compatibility 1-3

Understanding the Subnet Manager 1-4

Understanding Secure Shell 1-4

CHAPTER 2

FACT Quick Configuration 2-1

Obtaining and Installing FACT Software 2-1

Obtaining FACT Software 2-1

Installing FACT Software 2-2

Configuring FACT Software 2-2

Creating the Master Configuration File 2-2

Creating the Credentials File 2-3

Troubleshooting FACT Configuration 2-4

CHAPTER 3

Installing and Configuring FACT 3-1

FACT Installation and Configuration Checklist 3-1

Obtaining and Installing FACT Software 3-2

Obtaining the FACT RPM File 3-2

Installing FACT 3-2

Obtaining and Installing ibspark 3-2

- About FACT Configuration Files 3-3
 - About Master Configuration Files 3-3
 - About Managed Node Definitions Files 3-4
 - About Credentials Files 3-5
 - About GUID Name Definitions 3-9
- Configuring FACT 3-10
 - Creating a Master Configuration File 3-10
 - Creating a Separate Managed Node Definitions File (Optional) 3-11
 - Creating a GUID Name Definition File (Optional) 3-11
 - Creating a Credentials File 3-12

CHAPTER 4

- Using FACT 4-1**
 - Scanning 4-1
 - FACT Scanning Commands 4-2
 - Maintaining the Repository 4-2
 - Repository Management Commands 4-3
 - Showing Network Information 4-4
 - Using Annotation 4-4
 - Using Port Counters 4-5
 - Port Counters Commands 4-6
 - Installing Firmware with FACT 4-6
 - Installing Firmware in Managed Switches 4-6
 - Installing Firmware on Unmanaged Switches 4-7

CHAPTER 5

- FACT Command Reference 5-1**
 - Using the CLI 5-1
 - Using FACT Command Arguments 5-1
 - Using Command Modes 5-4
 - Command-Line Interface 5-5
 - Command-Line Options 5-5
 - Environment Variables 5-7
 - Correcting Commands 5-8
 - Editing the CLI 5-8
 - Commands 5-10
 - Command Groups 5-53

INDEX



Preface

This preface describes who should read the *Cisco Fabric Analysis and Correlation Tool Kit User Guide*, how it is organized, and its document conventions.

This chapter includes the following sections:

- [Audience, page v](#)
- [Organization, page v](#)
- [Related Documentation, page vi](#)
- [Conventions, page vi](#)
- [Obtaining Documentation and Submitting a Service Request, page vii](#)

Audience

This guide is for experienced network administrators who are responsible for configuring and managing Cisco Server Fabric Switches, InfiniBand hosts, Fibre Channel gateways, Ethernet gateways, and Subnet Managers.

Organization

This guide is organized as follows:

Chapter	Title	Description
Chapter 1	Understanding the Fabric Analysis and Correlation Toolkit	Introduces FACT, its key terms and concepts, and its hardware and software compatibility.
Chapter 2	FACT Quick Configuration	Provides the minimal tasks for obtaining, installing, and configuring FACT.
Chapter 3	Installing and Configuring FACT	Provides details about how to obtain, install, and configure FACT software.
Chapter 4	Using FACT	Describes how to use FACT to solve common IB management problems.

Chapter 5	FACT Command Reference	Describes the command-line interface and shows all commands used in FACT.
Appendix A	Acronyms and Abbreviations	Expands the acronyms and abbreviations that are used in this guide.

Related Documentation

For additional information about Cisco products related to Cisco Server Fabric Switches and software configuration, see the following publications:

- *Cisco SFS 7024 InfiniBand Server Switch Hardware User Guide*
- *Cisco SFS 7024 InfiniBand Server Switch Installation and Configuration Note*
- *Cisco SFS 7012 InfiniBand Server Switch Hardware User Guide*
- *Cisco SFS 7012 InfiniBand Server Switch Installation and Configuration Note*
- *Cisco SFS 7008P InfiniBand Server Switch Hardware Installation Guide*
- *Cisco SFS 7008P InfiniBand Server Switch Installation and Configuration Note*
- *Cisco SFS 70008 InfiniBand Server Switch Hardware Installation Guide*
- *Cisco SFS 7008 InfiniBand Server Switch Quick Start Guide*
- *Cisco SFS 7000P and SFS 7000D InfiniBand Server Switches Hardware Installation Guide*
- *Cisco SFS 7000 Switch Hardware Installation Guide*
- *Cisco SFS 7000 InfiniBand Server Switch Quick Start Guide*
- *Cisco SFS InfiniBand Fibre Channel Gateway User Guide*
- *Cisco SFS InfiniBand Ethernet Gateway User Guide*
- *Cisco SFS InfiniBand Redundancy Configuration Guide*
- *Cisco SFS Product Family Element Manager User Guide*
- *Cisco SFS Product Family Chassis Manager User Guide*
- *Cisco SFS Product Family Command Reference*
- *Cisco InfiniBand Host Channel Adapter Hardware Installation Guide*
- *Cisco SFS InfiniBand Host Drivers User Guide for Linux*
- *InfiniBand Hardware Installation and Cabling Guide*

Conventions

This document uses the following conventions:

Convention	Description
boldface font	Commands, command options, and keywords are in boldface . Bold text indicates text that you must enter as-is.
<i>italic</i> font	Arguments in commands for which you supply values are in <i>italics</i> . Italics not used in commands indicate emphasis.

Convention	Description
Menu1 > Menu2 > Item...	Series indicate a pop-up menu sequence to open a form or execute a desired function.
[]	Elements in square brackets are optional.
{ x y z }	Alternative keywords are grouped in braces and separated by vertical bars. Braces can also be used to group keywords and/or arguments; for example, { interface <i>interface</i> type }.
[x y z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
screen font	Terminal sessions and information the system displays are in screen font.
boldface screen font	Information you must enter is in boldface screen font .
<i>italic screen font</i>	Arguments for which you supply values are in <i>italic screen font</i> .
^	The symbol ^ represents the key labeled Control—for example, the key combination ^D in a screen display means hold down the Control key while you press the D key.
< >	Nonprinting characters, such as passwords are in angle brackets.
[]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

<http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html>

Subscribe to the *What's New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS version 2.0.



CHAPTER 1

Understanding the Fabric Analysis and Correlation Toolkit

This chapter describes FACT, provides information about supported hardware and software, and includes the following sections:

- [About FACT, page 1-1](#)
- [How FACT Works, page 1-2](#)
- [Hardware and Software Compatibility, page 1-3](#)
- [Understanding the Subnet Manager, page 1-4](#)
- [Understanding Secure Shell, page 1-4](#)

About FACT

The Cisco Fabric Analysis and Correlation Toolkit (FACT) is a software application that aids in bringing up and debugging InfiniBand networks. FACT analyzes hardware objects and identifies components in the fabric that need attention, such as faulty cables, loose cables, faulty HCAs, and faulty switches and switching modules. FACT also analyzes InfiniBand networks at a high level to detect errors in the fabric by building coherent models of the relationships between InfiniBand objects (nodes, ports, GUIDs, and LIDs) and physical objects (chassis, cards, and ports).

FACT collects and reports the network topology and connectivity at the level of a simple subnet, monitors the logs on Cisco SFS switches, and summarizes and annotates the most important information therein. FACT maintains a historical view of a subnet and can show how the topology changes over time.

FACT also does the following:

- Provides tools to explore the relationships between InfiniBand objects and physical objects
- Enables you to annotate logs, especially Subnet Manager logs, turning each GUID into a meaningful location in a chassis
- Collects InfiniBand port counters and displays them, also with meaningful locations
- Updates firmware over a number of switches
- Enables and disables ports
- Sets port speed and width

This toolkit aids system administrators in managing and deploying a new IB fabric. FACT provides an abstraction layer across the Cisco IB switching family.

How FACT Works

- [Scanning and Reporting, page 1-2](#)
- [Repository, page 1-2](#)
- [Connections, page 1-3](#)
- [Network Maintenance, page 1-3](#)

Scanning and Reporting

FACT scans a subnet by logging in to the management ports of many switches, finding the master Subnet Manager (or the host Subnet Manager if the host is listed as a managed node), and interrogating it through the CLI. FACT then constructs a view of the network topology and saves it to a file to answer later queries. If presented with a network with no Subnet Managers or with several master Subnet Managers, FACT has reduced function, but it can report the location of the master and standby Subnet Managers. For more information about the Subnet Manager, see the [“Understanding the Subnet Manager”](#) section.

FACT has a set of queries to scan the fabric, similar to those in the Subnet Manager, where it can show which switches are in the network, which ports are active, how they are connected, and so on. By reporting this information, FACT differentiates itself from the Subnet Manager **show** commands because it reports both at the chip level (InfiniBand nodes and ports) and at the chassis level (chassis, slots, and external ports).

Along with scanning the fabric for information about connectivity, FACT can perform a thorough technical-support scan, in which it collects voluminous diagnostic information from every SFS OS switch, collecting information that Cisco TAC or engineering can use to diagnose problems remotely. (For more information about scans, see the [“Scanning”](#) section.)

Repository

All information that FACT collects is stored in its repository, which is a directory in the file system. The FACT repository contains the following items:

- Results of all scans
- A log of analysis errors
- Transcripts of all switch CLI sessions
- A “last-known neighbor” map used to remember neighbor relationships after links go down
- A pointer to the current scan

The repository also maintains the notion of a “current scan.” Because queries are always performed against the current scan, the repository has a history mechanism that allows the current scan to be rolled back to an earlier version. (For more information about the repository, see the [“Maintaining the Repository”](#) section.)

Connections

The scanning function requires that FACT be able to log in to each switch in the network. Also, if you are using the Cisco High-Performance Subnet Manager (HSM), FACT must be able to log in to each host that is running HSM.

FACT has a variety of ways to connect:

- Using SSH to connect to a switch management port
- Using SSH to log in to a host running HSM and starting the HSM CLI
- Using SSH to log in to a host running HSM and running the HSM CLI directly as a subprocess if HSM and FACT are on the same host

The credential files control which mechanisms FACT uses for each managed node. For more information about credentials files, see the [“About Credentials Files”](#) section.

Network Maintenance

FACT can query the firmware versions on switches in a network and perform firmware updates, either on individual switches or on multiple switches simultaneously.

FACT can also control ports. FACT can enable or disable one or several ports, individually, and change their width and speed. Using this port-control capability is more desirable than using the existing switch CLI because FACT is a single point of control. FACT allows the port to be specified by either the chassis or the chip location.

Hardware and Software Compatibility

FACT runs on a Linux host and requires the following supported software distributions:

- RedHat Enterprise Linux, Version 4 or 5, or SUSE Linux Enterprise Server Distribution, Version 9 or 10
- Python, Version 2.3 or later

FACT can log in to Cisco Server Fabric Switches that run Cisco SFS OS, to OEM switches (Cisco SFS 7012 and Cisco SFS 7024), and to Unix/Linux-based hosts, including those that run the host-based Subnet Manager. FACT can also monitor and control unmanaged switches to a limited extent by using in-band InfiniBand messages.

FACT uses the `ibspark` tool to upgrade firmware on unmanaged switches. FACT uses the `ibportstate` tool to control ports. These tools must be installed and available on a Linux host that is directly connected to the InfiniBand network and that FACT can log in to through SSH. FACT performs all other operations through IP to the management ports on a device, so it can run on any network-connected host.

FACT can optionally use configurations that map node GUIDs and system image GUIDs to user-specified names. This option is especially useful when working with unmanaged switches because unmanaged switches do not have IP addresses or other identifiers; their only identifiers are their node GUIDs. However, GUID names can be used for any switch or channel adapter.

Understanding the Subnet Manager

FACT works closely with the Subnet Manager to understand the network fabric because the Subnet Manager configures and maintains fabric operations. The Subnet Manager is the central repository of all information that is required to set up and bring up the InfiniBand fabric.

The master Subnet Manager does the following:

- Discovers the fabric topology
- Discovers end nodes
- Configures switches and end nodes with their parameters, such as the following:
 - LIDs
 - GUIDs
 - Partition Keys (P_Keys)
- Configures switch forwarding tables
- Receives traps from Subnet Management Agents (SMAs)
- Sweeps the subnet, discovering topology changes and managing changes as nodes are added and deleted

A network may contain multiple Subnet Managers acting as standbys, but it may contain only one master Subnet Manager.

Understanding Secure Shell

Secure Shell (SSH) is a network protocol that provides a secure remote access connection to network devices. Cisco FACT uses SSH to provide secure communication from network computers to the Cisco SFS OS CLI.



CHAPTER 2

FACT Quick Configuration

This chapter includes the minimal tasks involved in installing and configuring FACT. For details about these installation and configuration tasks, see [Chapter 3, “Installing and Configuring FACT.”](#)

This chapter includes the following sections:

- [Obtaining and Installing FACT Software, page 2-1](#)
- [Configuring FACT Software, page 2-2](#)
- [Troubleshooting FACT Configuration, page 2-4](#)

Obtaining and Installing FACT Software

This section includes the following topics:

- [Obtaining FACT Software, page 2-1](#)
- [Installing FACT Software, page 2-2](#)



Note

Before you obtain and install FACT, ensure that you are running the following software on your system: RedHat Enterprise Linux, Version 4 or 5, or SUSE Linux Enterprise Distribution, Version 9 or 10, and Python, Version 2.3 or later.

Obtaining FACT Software

FACT is distributed as an RPM file. To obtain a copy of the FACT RPM, perform the following steps:

-
- Step 1** Navigate to the Cisco software download site at the following URL:
<http://www.cisco.com/kobayashi/sw-center/index.shtml>
- If you are not already a registered user, you must register at the above URL to gain access to software downloads.
- Step 2** Select the software product category by clicking the **Server Network and Virtualization** link.
- Step 3** Select a device by choosing **Server Fabric Software > Infiniband Management Software > Cisco Fabric Analysis and Correlation Toolkit**.
-

Installing FACT Software

To install the FACT software RPM, perform the following steps:

-
- Step 1** Log in to your console as a super user.
- Step 2** Install the RPM by entering the following command:
- ```
rpm -i cisco-fact-version-noarch.rpm
```
- Step 3** FACT installs the software without status messages. To verify that you have installed the correct version of FACT, enter the **fact -v** command, and verify the version number in the command output.
- 

## Configuring FACT Software

This section includes the following topics:

- [Creating the Master Configuration File, page 2-2](#)
- [Creating the Credentials File, page 2-3](#)

## Creating the Master Configuration File

For more information about master configuration files, see the [“About Master Configuration Files”](#) section.

To create the master configuration file, perform the following steps:

- 
- Step 1** Log in to your host as a super user.
- Step 2** Using a text editor, create a master configuration file named `.fact.conf` in your home directory.
- Step 3** Enter the following line into the configuration file:
- ```
credential-file: ~/.fact-credentials
```
- Step 4** For each managed switch that is running SFS OS in your network, add a line to the master configuration file that specifies the hostname or IP address:
- ```
managed-nodes: SFSOS switch name
```
- For information about managed nodes, see the [“About Managed Node Definitions Files”](#) section.
- Step 5** For each managed OEM switch that is in your network (either a Cisco SFS 7012 or a Cisco SFS 7024), add a line to the file that specifies the hostname or IP address:
- ```
managed-nodes: OEM switch name
```
- Step 6** If you are using the High-Performance Subnet Manager, specify each host that runs the High-Performance Subnet Manager in the master configuration file:
- ```
managed-nodes: host name
```

**Step 7** (Optional) FACT usually determines the names of hosts on the network without needing to log in to them. If FACT does not identify the hosts or if you want FACT to collect additional diagnostic information from the hosts, add each host in your network to the master configuration file:

```
managed-nodes: host name
```

**Step 8** (Optional) If you have unmanaged switches in your network, add a line to the master configuration file that specifies the GUID name and creates a name for the switch. (For more information about GUID names, see the [“About GUID Name Definitions”](#) section.) You may use any name for the switch, but you must use the following format for the line that you add to the file:

```
guid-name: 00:1b:0d:0b:00:00:ca:a6 switchA
guid-name: 00:1b:0d:0b:00:00:ca:aa switchB
```

---

## Creating the Credentials File

For more information about the credentials file, see the [“About Credentials Files”](#) section.

To create a credentials file, perform the following steps:

---

**Step 1** Using a text editor, create a file named `.fact-credentials` in your home directory.

**Step 2** For each managed node (switch or host) that you listed in the master configuration file, add a username and a password that FACT can use to log in to that managed node. Choose from Case 1 or Case 2 in the following steps:

- a. **Case 1**—If each switch or host in your network has a different username or password, create a separate stanza in the credentials file for each switch and each host, as shown in the following example:

```
SFSOS switch name
 user: username
 password: password
```

```
OEM switch name
 host myhost[1-8]
 user: username
 password: password
```

```
host name
 user: username
 password: password
```

- b. **Case 2**—If all managed nodes of a given type (host, SFS OS switch, or OEM switch) have the same username and password, create one stanza for each node type, as shown in the following examples:

```
SFSOS switch *
 user: username
 password: password
```

```
OEM switch *
 user: username
 password: password
```

```
host *
 user: username
 password: password
```

- Step 3** Save the credentials file that you just created.
- Step 4** Verify the configuration by entering the **fact scan fabric** command, which scans the managed nodes in the main configuration file.
- Step 5** If FACT reports errors while scanning, see the [“Troubleshooting FACT Configuration”](#) section.
- 

## Troubleshooting FACT Configuration

If FACT reports errors while scanning, consult this section for common symptoms, possible causes, and recommended actions.

**Symptom** FACT returns an error message indicating that permission is denied: “*name*: permission denied.”

**Possible Cause** The credentials file is either missing or incorrect.

**Recommended Action** Correct the credentials file. (See the [“Creating the Credentials File”](#) section.)

**Symptom** FACT returns a time-out message: “ScanError: timeout scanning *name*.”

**Possible Cause** The master configuration file has an incorrect hostname or IP address, or the host is not reachable using the name that appears in the file.

**Recommended Action** Verify that you have entered the correct hostname or IP address in the master configuration file. (See the [“Creating the Master Configuration File”](#) section.) Also, verify that the managed node is reachable through SSH.

**Symptom** When you attempt to scan the network, FACT returns a message saying that multiple master Subnet Managers are found: “ScanError: multiple Master Subnet Managers found.”

**Possible Cause** The master configuration file contains managed nodes from multiple subnets. FACT can scan a single subnet only.

**Recommended Action** Enter the **fact show subnet-managers** command to see all of the master and standby Subnet Managers that FACT found. From the master configuration file and from the credentials file, remove the Subnet Managers that are not in the subnet that you are scanning.

**Symptom** FACT cannot find a master Subnet Manager: “ScanError: no Master Subnet Manager found.”

**Possible Cause** The node that is running the master Subnet Manager is not in the master configuration file. (See the [“Creating the Master Configuration File”](#) section.)

**Recommended Action** Check the master configuration file, and add the node that is running the master Subnet Manager if it does not exist in the file.





## CHAPTER 3

# Installing and Configuring FACT

---

This chapter describes how to install and configure FACT, and it includes the following sections:

- [FACT Installation and Configuration Checklist, page 3-1](#)
- [Obtaining and Installing FACT Software, page 3-2](#)
- [About FACT Configuration Files, page 3-3](#)
- [Configuring FACT, page 3-10](#)

## FACT Installation and Configuration Checklist

This section provides a checklist of the high-level tasks to follow as you obtain, install, and configure FACT. We suggest that you perform the tasks in the following order:

- Verify that your system is running the required software.  
(See the [“Hardware and Software Compatibility”](#) section.)
- Obtain and install the FACT RPM file.  
(See the [“Obtaining the FACT RPM File”](#) and [“Installing FACT”](#) sections.)
- (Optional) Obtain and install the optional software tools.  
(See the [“Obtaining and Installing ibspark”](#) section.)
- Create the master configuration file.  
(See the [“About Master Configuration Files”](#) and [“Creating a Master Configuration File”](#) sections.)
- (Optional) Create the managed-node definitions files.  
(See the [“About Managed Node Definitions Files”](#) and [“Creating a Separate Managed Node Definitions File \(Optional\)”](#) sections.)
- Create the credentials files.  
(See the [“About Credentials Files”](#) and [“Creating a Credentials File”](#) sections.)
- (Optional) Create the GUID name definitions.  
(See the [“About GUID Name Definitions”](#) and [“Creating a GUID Name Definition File \(Optional\)”](#) sections.)

After you have installed FACT and configured all required and optional setups for your network, you can begin to scan the network with FACT by using the information in the [Chapter 4, “Using FACT.”](#)

# Obtaining and Installing FACT Software

This section includes the following topics:

- [Obtaining the FACT RPM File, page 3-2](#)
- [Installing FACT, page 3-2](#)
- [Obtaining and Installing ibspark, page 3-2](#)

## Obtaining the FACT RPM File

FACT is distributed as an RPM file. To obtain a copy of the FACT RPM, perform the following steps:

- 
- Step 1** Navigate to the Cisco software download site at the following URL:  
<http://www.cisco.com/kobayashi/sw-center/index.shtml>
- If you are not already a registered user, you must register at the preceding URL to gain access to software downloads.
- Step 2** Select the software product category by clicking the **Server Network and Virtualization** link.
- Step 3** Select a device by choosing **Server Fabric Software > InfiniBand Management Software > Cisco Fabric Analysis and Correlation Toolkit**.
- 

## Installing FACT

To install the RPM, perform the following steps:

- 
- Step 1** Log in as a super user.
- Step 2** Install the FACT RPM file by entering the following command:
- ```
# rpm -i cisco-fact-version-noarch.rpm
```

After FACT installs you can obtain and install additional software to make use of the full FACT functionality. (See the “[Obtaining and Installing ibspark](#)” section.) Otherwise, you must create the FACT configuration files.

Obtaining and Installing ibspark

FACT enables users to upgrade firmware on unmanaged switches and on the control ports. To upgrade firmware on unmanaged switches you must obtain and install the ibspark tool.



Note

To control the ports you must have the `ibportstate` utility available on a host with which FACT can communicate. The `ibportstate` utility is an Open Fabrics utility that is not included with FACT software.

To obtain and install the optional ibspark software tool, perform the following steps:

-
- Step 1** Navigate to the Cisco software download site at the following URL:
<http://www.cisco.com/kobayashi/sw-center/index.shtml>
- If you are not already a registered user, you must register at the preceding URL to gain access to software downloads.
- Step 2** Select the software product category by clicking the **Server Network and Virtualization** link.
- Step 3** Select the ibspark tool by choosing **Server Fabric Software > InfiniBand Management Software > Cisco Fabric Analysis and Correlation Toolkit > ibspark**.
- Step 4** Install ibspark.
-

About FACT Configuration Files

FACT uses four types of configuration files that you must create. These configuration files provide information about the network that you want FACT to analyze. You create these files after you install the FACT RPM (see “[Installing FACT](#)”) and before you use FACT. After familiarizing yourself with the configuration file types in this section, proceed to the “[Configuring FACT](#)” section to create your FACT configuration files.

This section includes the following topics:

- [About Master Configuration Files, page 3-3](#)
- [About Managed Node Definitions Files, page 3-4](#)
- [About Credentials Files, page 3-5](#)
- [About GUID Name Definitions, page 3-9](#)



Note

FACT can operate with no configuration; however, FACT is minimally useful in an unconfigured state.

About Master Configuration Files

FACT reads a master configuration file, which may specify other subsidiary configuration files. FACT searches for the master configuration file in the following files, using the first master configuration file that it finds:

```
file specified by -c or --config-file command option
file specified by FACT_CONF environment variable
~/fact.conf (i.e., .fact.conf in user's home directory)
/etc/fact.conf
compiled-in defaults
```

The master configuration file contains name-value pairs, one per line. The parameter is separated from its value by a colon. Blank lines and comments that start with a pound character (#) are ignored.

The following parameters are permitted:

- repository: *directory*
 The directory where FACT stores its data.

- `credential-file: file`
A file containing credentials. (See the [“About Credentials Files”](#) section.)
- `managed-nodes: type pattern`
A managed-node definition. (See the [“About Managed Node Definitions Files”](#) section.)
- `managed-node-file: file`
A file containing managed node definitions. (See the [“About Managed Node Definitions Files”](#) section.)
- `guid-name: guid name`
A GUID name definition. (See the [“About GUID Name Definitions”](#) section.)
- `guid-names-file: file`
A file containing GUID name definitions. (See the [“About GUID Name Definitions”](#) section.)
- `log-file: file`
A log file created by the syslog and monitored by FACT.
- `subnet-manager: domain name or IP address`
The name of the Subnet Manager that FACT should use.
- `ibportstate-host: host`
The host on which FACT should run `ibportstate` to control switch ports. The default is `localhost`.
- `ibportstate-command: path`
The path to the **ibportstate** command on the `ibportstate` host. The default is `ibportstate`.
- `ibspark-host: host`
The host on which FACT should run `ibspark`. The default is `ibspark`.
- `ibspark-command: path`
The path to the **ibspark** command on the `ibspark` host. The default is `localhost`.

**Note**

The `credential-file`, `log-file`, `managed-nodes`, `managed-node-file`, `guid-name`, and `guid-names-file` parameters may be repeated any number of times, and their effects are cumulative.

The following example shows one possible master configuration file:

```
repository: /var/local/db/fact
credential-file: /etc/fact/cred
```

For information about creating master configuration files, see the [“Creating a Master Configuration File”](#) section.

About Managed Node Definitions Files

A managed node is either a host or a managed switch. FACT must know about all of the managed nodes in any given network so that it can scan the network. Managed node definitions tell FACT the hostname or IP address and the type of each managed node. If managed nodes are not defined, FACT cannot connect to the switch management ports and collect the necessary information.

Names may be specified using node-list expansions, which are separated by commas, with no spaces. FACT has three types of expansions:

- Alternate—For example, leaf[A,C,D] expands to leafA, leafC, leafD.
- Range—For example, leaf[1-10] expands to leaf1, leaf2...leaf10.
- Combination of alternate and range—For example, rack[A,C]host[1-32] expands to rackAhost1, rackAhost1...rackChost32 (64 names total).

FACT managed-nodes have three types:

- SFSOS switch—Any Cisco Server Fabric Switch running Cisco SFS operating system software
- OEM switch—The Cisco SFS 7012 and Cisco SFS 7024
- host—A host running Unix

Managed-nodes can be listed directly in the master configuration file using the "managed-nodes" directive or in a separate file. A separate managed-nodes file has a node type and a list, as shown in the following example:

```
SFSOS switch switch[00-07]
host host-r[1-3,5]-[0-32]
```

The ranges in square brackets are expanded.

FACT looks in the following places for managed node definitions, with the highest priority sources listed first:

- Command options—managed-nodes=<type>:<pattern>
- Files specified by—managed-nodes-from=<file> options
- Definitions in the FACT_MANAGED_NODES environment variable
- Files in FACT_MANAGED_NODE_FILES environment variable (colon-separated list)
- Definitions in managed-nodes parameters in the master configuration file
- Files in managed-node-file parameters in the master configuration file

The managed node definitions file expands to six switch names: rackAleaf1 through rackAleaf3 and rackCleaf1 through rackCleaf3. It also expands to 64 host names: rackBcompute01 through rackDcompute32. If your managed-nodes follow this type of simple name scheme, you may place them into a master configuration file, as shown in the following example with managed-node parameters in the master configuration file:

```
managed-nodes: SFSOS switch rack[A,C]leaf[1-3]
managed-nodes: host rack[B,D]compute[01-32]
```

For information about creating a managed node definitions file, see the [“Creating a Separate Managed Node Definitions File \(Optional\)”](#) section.

About Credentials Files

This section includes the following topics:

- [Device Types, page 3-6](#)
- [Wildcard Values, page 3-7](#)
- [Name-Value Pairs, page 3-7](#)

For each managed node into which it logs, FACT must know which username and password to use. The credentials file provides this information.

A credentials file consists of multiple stanzas. Each stanza begins with a device type line and is followed by several name-value pairs. Blank lines and comments starting with the pound sign (#) are ignored.

FACT looks in the following locations for credential files. The highest-priority locations, which are listed first, override definitions in later locations.

- Command options—`credentials-from=file`
- Files specified in `FACT_CREDENTIAL_FILES` environment variable (colon-separated list)
- Files specified in `credentials-file` parameters in the master configuration file
- Built-in defaults

FACT contains the following built-in default credentials:

SFS OS switch

method: ssh

user: super

password: super

OEM switch

method: ssh

user: admin

password: admin

host

method: ssh

HSM-command: `sudo /usr/local/topspin/sbin/ib_sm_cli || sudo /usr/sbin/ib_sm_cli`

vstat command: `/usr/local/topspin/bin/vstat --verbose`

password: no default password exists

For information about creating a credentials file, see the [“Creating a Credentials File”](#) section.

Device Types

The device type line in the credentials file contains a device type, followed by a wildcard value that matches a set of device names. [Table 3-1](#) lists the device types that may appear in the device type line, and [Table 3-2](#) lists the wildcard values.

Table 3-1 Device Types

Device Type	Description
host	A host running Unix
SFSOS switch	A switch running SFS OS
OEM switch	A Cisco OEM switch (SFS 7012 and SFS 7024 only)

Wildcard Values

A wildcard is matched against managed-node names using a specific set of values. [Table 3-2](#) lists the wildcard values against which the managed-node names are matched.



Note

In the wildcard values represented below, where letters are used, actual wildcard can be either letters or numbers.

Table 3-2 Wildcard Values

Wildcard Value	Description
*	Matches any substring
?	Matches any single character
[a,b]	Matches either a or b
[a-b]	Matches anything in the range of a through b, where a and b can be either letters or numbers
[a,c-e,]	Matches either a, c, d, or e

Name-Value Pairs

The credentials file must contain name-value pairs that have specific, allowed parameters.

[Table 3-3](#) lists the legal parameters for the name-value pairs in the credentials file.

Table 3-3 Parameters for Name-Value Pairs

Name-Value Pair	Description
user	Login username. The default is “super” for SFS OS switches and “admin” on OEM switches. No default exists on the host. Most users do use this parameter.
password	Login password. The default is “super” on SFS OS switches and “admin” on OEM switches. No default exists on other devices. Most users do use this parameter.
method	Method in which FACT connects to the device. Legal values are “SSH” and “direct.” The default is “direct” for the localhost and “SSH” for others. Most users do not use this parameter.
port	TCP port number to use for SSH. The default is 22. Most users do not use this parameter.
ssh-identity	The SSH identity file to use for authentication. FACT does not use a default identity file, yet SSH defaults to ~/.ssh/id_rsa or id_dsa. Most users do not use this parameter.

Table 3-3 Parameters for Name-Value Pairs (continued)

Name-Value Pair	Description
HSM-command	The executable program to invoke the HSM CLI. The default is /usr/local/topspin/sbin/ib_sm_cli /usr/sbin/ib_sm_cli. Most users do not use this parameter.
vstat-command	The vstat command, used to invoke vstat, includes the full path to the command and the verbose argument, which returns more information. The default is /usr/local/topspin/bin/vstat -- verbose.  Note The vstat parameter is part of the Cisco host driver stack. If you are using OFED host drivers and your Cisco stack is installed in the normal way, you do not need to use the verbose parameter. Most users do not use this parameter.

**Note**

The SSH method uses secure shell (SSH) to connect to the managed node. The direct method is only used to scan the host upon which FACT is running. (See the “[Understanding Secure Shell](#)” section for more information about SSH.)

The following example shows a possible credentials file:

```
# This is a comment.
SFSOS switch switch4*
  user      : username
  password: t0psekrlt
host hsm-[1-3]
  method:      ssh
  user:        fact
  ssh-identity: ~/.ssh/id_rsa
```

About GUID Name Definitions

A GUID is a 64-bit number that is used to identify several types of InfiniBand components. Components that have GUIDs are Host Channel Adapters, switch chips, ports, and switch chassis. GUIDs are usually displayed as hexadecimal octets separated by colons, as shown in the following example:

```
00:02:c9:02:00:22:2a:c4
```

FACT uses GUID name definitions to associate GUIDs with names. FACT automatically builds its own associations between GUIDs and names. If FACT knows the host domain name or IP address, or the switch management port domain name or IP address, it uses this information as the switch name. If FACT does not know the GUID name of a switch, FACT refers to the switch by its system image GUID.

Optionally, you can augment and override the user-generated list and assign any name that you choose to a GUID by creating a GUID name definitions file in the master configuration file. FACT uses that GUID name when referring to that object and when showing information from a scan. Assigned GUID name definitions are most useful when you work with unmanaged switches. Without an assigned GUID name definition, there is no way for FACT to refer to an unmanaged switch except by its node GUID.

FACT looks in the following locations for GUID name definitions. Higher-priority sources are listed first in the following list, and they override lower priority sources:

- Command options—`guid-name=guid:name`
- Files specified by—`guid-names-from=file` command options
- Definitions in the `FACT_GUID_NAMES` environment variable (comma-separated list)
- Files in `FACT_GUID_NAME_FILES` environment variable (colon-separated list)
- Definitions in `guid-name` parameters in the master configuration file
- Files in `guid-names-file` parameters in the master configuration file

The GUID name definition contains two pieces of information, the eight-byte GUID name and the name you assign to it, separated by spaces or commas. The following example shows a possible name definition file:

```
00:11:22:33:44:55:66:77 myswitch01
00:11:22:33:44:55:66:78 myswitch02
00:11:22:33:44:55:66:79 myswitch03
11:22:33:44:55:66:77:88 myhost01
```

If you do not want FACT to log in to each host in your cluster, and if FACT cannot determine the hostnames from the Subnet Manager, you can use GUID names to help FACT display useful names for your hosts.

**Note**

You can also attach a name to a system image GUID, a chassis GUID, or a port GUID.

For information about creating GUID name definitions files, see the [“Creating a GUID Name Definition File \(Optional\)”](#) section.

Configuring FACT

For FACT to fully analyze a network you must configure the following files, which provide FACT with information about that network fabric.

FACT has four types of configurations for which you must set up a configuration file in the following order:

- [Creating a Master Configuration File, page 3-10](#)
- [Creating a Separate Managed Node Definitions File \(Optional\), page 3-11](#)
- [Creating a GUID Name Definition File \(Optional\), page 3-11](#)
- [Creating a Credentials File, page 3-12](#)



Note

FACT can operate with no configuration; however, the application is minimally useful in an unconfigured state. To use FACT without configuration, you must run FACT on the same host as the HSM and as the super user.

Creating a Master Configuration File

The master configuration file tells FACT how to get started, how to find the repository, how to find the credentials files, and how to find the necessary information that it needs to connect with the network that it must analyze. For information about master configuration files, see the [“About Master Configuration Files”](#) section.

To create the master configuration file, perform the following steps:

-
- Step 1** Using a text editor, create the master configuration file in your home directory. Begin by entering the following line into the file:
- ```
credential-file: ~/.fact-credentials
```
- Step 2** To define managed-nodes so that FACT can connect to the management ports and collect the necessary information from the switches in the fabric, add lines to the file that specify the DNS names or IP addresses of each of your managed IB switches. Optionally, specify the names of each of your hosts. If you are using the HSM, you must specify that host in the master configuration file, called .fact.conf.
- The lines for a Cisco SFS Server Switch running SFS OS appear as the following:
 

```
managed-nodes: SFSOS switch
```
  - The lines for a Cisco SFS 7012 or Cisco SFS 7024 Server Switches appear as the following:
 

```
managed-nodes: OEM switch
```
  - The lines for a host appear as the following:
 

```
managed-nodes: host
```

You can use a node list expansion for the name if several devices have similar names. For example, if you have a small network with eight hosts, a Cisco SFS 3001 Server Switch, and a Cisco SFS 7012 Server Switch, and their DNS names are myhost1 through myhost8, my3001, and my7012, respectively, you would create the following master configuration file:

```
credentials-file: ~/.fact-credentials
managed-nodes: host myhost[1-8]
```

```
managed-nodes: SFSOS switch my3001
managed-nodes: OEM switch my7012
```

Step 3 Save the file.

---

## Creating a Separate Managed Node Definitions File (Optional)

If your configuration is too large or complex to describe inline in the master configuration file, you can create a separate managed nodes definitions file. (For more information about managed node definitions, see the [“About Managed Node Definitions Files”](#) section.)

To create a managed node definitions file, perform the following steps:

Step 1 Using a text editor, create the managed nodes definitions file in your home directory.

Step 2 For each managed switch that is running SFS OS in your network, add a line to the managed node definition file that specifies the name or IP address:

```
SFSOS switch name
```

Step 3 For each OEM-managed switch that is in your network, add a line to managed node definition file that specifies the name or IP address:

```
OEM switch name
```

Step 4 (Optional) If you do not want FACT to collect additional diagnostic information from the hosts, add a line to the managed node definition file that specifies the hostname or IP address:

```
host name
```

Step 5 Add the managed node definition file to the master configuration file:

```
managed-node-file: ~/mnode-definition
```

Step 6 Save the file.

---

## Creating a GUID Name Definition File (Optional)

For easy identification, you may choose to assign any name to a GUID. If you choose not to assign names, GUIDs may still be identified by their numbers. For more information about GUID name definitions, see the [“About GUID Name Definitions”](#) section.

To create a GUID name definitions file and assign names to GUIDs, perform the following steps:

Step 1 Using a text editor, create a file.

Step 2 For each GUID, create a name definition that contains the following two pieces of information, separated by a space:

- a. the eight-byte GUID
- b. the name you choose to assign it

The following example shows a possible name definition file:

```
00:11:22:33:44:55:66:77 switch01
00:11:22:33:44:55:66:78 switch02
00:11:22:33:44:55:66:79 switch03
11:22:33:44:55:66:77:88 host01
```

**Step 3** Save the file.

---

## Creating a Credentials File

A credentials file is the place in which you store the information that tells FACT how to log in to other machines or managed-nodes. You create a username and password for each node into which FACT can log in. For more information about credentials files, see the [“About Credentials Files”](#) section.

To create a credentials file, perform the following steps:

---

**Step 1** Create a file named `.fact-credentials` in your home directory using a text editor.

**Step 2** For each managed node (switch or host) that you listed in the master configuration file that you previously created, called `fact.conf`, add a username and a password that FACT can use to log in to that managed node. For Cisco switches that use SFS OS, use "super" as both the default username and password; for OEM switches, which are the Cisco SFS 7012 and Cisco SFS 7024, use "admin" as both the default username and password, as shown in the following examples:

Cisco SFS OS switches

```
user: super
password: super
```

OEM switches

```
user: admin
password: admin
```

**Step 3** (Optional) You can use wildcards if several devices use the same username and password.

```
host myhost[1-8]
user: myname
password: secret
```

For more information about wildcards, see the [“Wildcard Values”](#) section.

**Step 4** Save the file.

---



# CHAPTER 4

## Using FACT

---

This chapter describes how to use FACT software and includes the following sections:

- [Scanning, page 4-1](#)
- [Maintaining the Repository, page 4-2](#)
- [Showing Network Information, page 4-4](#)
- [Using Annotation, page 4-4](#)
- [Using Port Counters, page 4-5](#)
- [Installing Firmware with FACT, page 4-6](#)

## Scanning

Before FACT can perform other network tasks, you must first scan the network to determine which hosts and switches are in the network and to determine how they are connected. FACT scans an entire subnet at the same time, connecting to each managed Cisco switch and, optionally, to each port.

FACT has four scan types:

- **Fabric scan**—Finds the static configuration of the network. After collecting a fabric scan, FACT can answer queries about the network components and determine how they are connected.
- **Tech-support scan**—Collects information that can be forwarded to a customer-support organization. A tech-support scan includes a fabric scan, which FACT can use, although FACT does not use the additional information. FACT saves that information for an expert to inspect.
- **Port-counters scan**—Collects port counters from the Performance Manager, in addition to the data collected in a fabric scan. See the Port Counters section for more information.
- **Incomplete scan**—A scan that has failed, either because of misconfiguration or because devices are not responding. The most common reason for an incomplete scan is either because the configuration does not include the device that is running the master Subnet Manager or because it includes two master Subnet Managers. In either case, change your configuration so that it includes exactly one master Subnet Manager. Alternately, you can force FACT to use the correct master Subnet Manager by setting `subnet-manager` in the master configuration file or with the `subnet-manager hostname` command option.



Note

---

If you are using the HSM, then you must allow FACT to scan the hosts running HSM.

---

You can use the **--trace** command option to show FACT communication with each managed node as it scans.

## FACT Scanning Commands

Use the following commands to scan with FACT:

- **scan fabric**
- **scan tech-support**

## Maintaining the Repository

FACT maintains a repository that contains the results of past scans. FACT considers the last-performed scan in the repository as the current scan, which it uses for queries, annotations, port control, and firmware upgrades. Whenever you scan the network, the new scan becomes the current scan, although the repository has a history mechanism that allows the current scan to be rolled back to an earlier version.

By default, the repository is located at `$HOME/.fact`.

Use the **show history** command to view all scans. Each scan has an index number that you use to select any particular scan. Use the **select** command to change the current scan.

The following example shows that the current scan is scan 6, as designated by the letter “Y”:

```
$ fact
fact> show history
index current type date subnet-manager

1 fabric 2007-12-27 19:37:25 PST ibmg-r1-120-1
2 fabric 2007-12-27 19:39:28 PST ibmg-r1-360-1
3 fabric 2008-01-03 17:09:51 PST kbob-s1
4 fabric 2008-01-03 20:14:50 PST 172.31.255.255
5 fabric 2008-01-08 15:30:29 PST kbob-s1
6 Y fabric 2008-01-08 15:33:52 PST kbob-s1
fact>
```

From the current scan shown in the previous example, the following example shows selecting scan 4:

```
fact> select 4
fact> show history
index current type date subnet-manager

1 fabric 2007-12-27 19:37:25 PST ibmg-r1-120-1
2 fabric 2007-12-27 19:39:28 PST ibmg-r1-360-1
3 fabric 2008-01-03 17:09:51 PST kbob-s1
4 Y fabric 2008-01-03 20:14:50 PST 172.31.255.255
5 fabric 2008-01-08 15:30:29 PST kbob-s1
6 fabric 2008-01-08 15:33:52 PST kbob-s1
fact>
```

You can delete old scans by selecting the scan number from the index and using the **delete** command. The following example shows deleting the first three scans from the list in the previous example:

```
$ fact
fact> delete scans 1-3
Proceed with scan delete [Y/n]? y
fact> show history
index current type date subnet-manager

 1 Y fabric 2008-01-03 20:14:50 PST 172.31.255.255
 2 fabric 2008-01-08 15:30:29 PST kbob-s1
 3 fabric 2008-01-08 15:33:52 PST kbob-s1
```

**Note**

After you deleted scans 1 through 3, the scans that were previously listed as scans 4, 5, and 6 have remained, but they have now moved up in the index to become scans 1, 2, and 3.

You can export the current scan to a file with the **export scan** command, or you can export the entire repository with the **export repository** command. Similarly, you can import a scan with the **import scan** command or the import the entire repository with the **import repository** command.

If you need assistance in troubleshooting your network, you can use the **export repository** command. You can export the repository, save it to your specified location, and then make a copy to send to technical support.

You can clean out your repository with the **delete scans** command.

**Caution**

Importing a repository deletes your existing repository and replaces it with the new repository that you import.

## Repository Management Commands

Use the following commands to maintain and manage your repository:

- **show history**
- **select**
- **export scan**
- **export repository**
- **import repository**
- **import scan**
- **delete scans**

## Showing Network Information

Use the FACT **help** command to list all commands that are available. Generally, each query has several different output formats. The common formats are summary, guides, and detail.

The notation "someswitch/M/N" is called a chassis-view port name. The notation shows port N on slot M of chassis someswitch. The notation "someswitch/L:M/N" is a chip-view port name. The notation shows port N on chip M of slot L of chassis someswitch. Some switches require a mapping between external port numbers and chip-level port numbers. The chassis view uses external port numbers, and the chip view uses internal port numbers.



### Note

The colon character (":") appears only in the chip view, so its appearance in a string indicates that the string is in chip view notation.

The **show ports** command shows the ports and also the port neighbors of each port. If a port has no current neighbor, the last-known neighbor is shown. Last-known neighbors appear in square brackets ("[...]").

## Using Annotation

When a file is annotated, FACT searches it for numeric constants and presents network components. It adds an annotation to each numeric constant, showing a human-readable name for that component.

FACT can annotate a log file or any file. When a file is annotated, FACT appends each GUID with a short explanation of what that GUID represents, as shown in the following example of an original log:

A port GUID is changed from this

```
00:00:00:00:00:00:00:00
```

to this:

```
00:00:00:00:00:00:00:00 (switch03/4/5, switch03/4:6/7, neighbor=switch12/2/3
```

The preceding example shows the port chassis view and chip view, followed by the neighbor chassis view and chip view.

A node GUID is changed to this:

```
00:00:00:00:00:00:00:00 (switch03/4:6)
```

A chassis GUID is changed to this:

```
00:00:00:00:00:00:00:00 (switch03)
```

A LID is changed to this:

```
23 (switch03/4/0, switch03/4:6/0, 00:00:00:00:00:00:00:00)
```

A port GUID is changed to this:

```
ff:12:60:1b:ff:ff:00:00:00:00:00:01:ff:33:2b:52 (switch03/4/0, switch03/4:6/0, neighbor=host14/1/1)
```

A Python source file, /usr/local/cisco/fact/annotation.py, contains sets of regular expression patterns for identifying interesting messages, for identifying messages containing node, chassis and port GUIDs and LIDs. Advanced users can edit that file to customize the syslog output.

You can use the annotation as an element in a UNIX shell pipeline to annotate GUIDs in any command output, as shown in the following example:

```
$ ibstat | fact annotate /dev/stdin
```

You can pass an entire syslog through FACT annotation, as shown in the following example:

```
$ fact annotate follow /var/logs/messages
```

If you do not provide a specific file with the annotate command, FACT looks for log-file parameters in the master configuration file.



Note

---

Annotation always uses the current scan.

---

## Using Port Counters

InfiniBand switch chips maintain a set of counters for each port. These counters count the occurrence of various types of errors. You can use FACT to collect and display port counters.

FACT can collect three types of port counter scans:

- **Raw counters**—The Subnet Manager does not need to be configured to collect raw port counters. When you collect raw port counters FACT tells the Subnet Manager to enter every switch chip in the subnet to collect its port counters.
- **Monitored**—The Subnet Manager must be configured to collect monitored port counters. The Subnet Manager can be configured to monitor specified counters or all counters.
- **Threshold**—The Subnet Manager must be configured to collect threshold port counters. The threshold scan collects just those monitored port counters that have exceeded their error thresholds.

For information about configuring the Subnet Manager, see the **ib pm** command in the *Cisco SFS Product Family Command Reference* at this URL:

[http://www.cisco.com/en/US/docs/server\\_nw\\_virtual/2.10.0\\_release/command/reference/cli210.html](http://www.cisco.com/en/US/docs/server_nw_virtual/2.10.0_release/command/reference/cli210.html)

For information about configuring the High-Performance Subnet Manager, see the **config pm monitored state** command and the **config pm threshold** command in the *Cisco High-Performance Subnet Manager User Guide* at this URL:

[http://www.cisco.com/en/US/products/ps6985/products\\_installation\\_and\\_configuration\\_guides\\_list.html](http://www.cisco.com/en/US/products/ps6985/products_installation_and_configuration_guides_list.html)

FACT can scan the port counters in a network. Viewing the port counters is a three-stage process:

1. Reset counters with the **reset port-counters** command. FACT records the reset in the scan history, which you can view later with the **show history** command.
2. Scan the counters (wait a minimum of 60 seconds or longer for errors to occur). For more information about the port-counters scan, see the “**Scanning**” section.
3. Show the port counters. To show the port counters or to restrict the output to those counters that are non-zero, use the **show counters** commands.

Consult the *Cisco High Performance Subnet Manager for InfiniBand Server Switches User Guide* or the *Cisco SFS Product Family Command Reference* for information about configuring port counter monitoring with the Cisco InfiniBand Subnet Manager.

## Port Counters Commands

Use the following commands to use port counters:

- [reset port-counters](#)
- [show counters monitored](#)
- [show counters raw](#)
- [show counters threshold](#)

## Installing Firmware with FACT

This section includes the following topics:

- [Installing Firmware in Managed Switches, page 4-6](#)
- [Installing Firmware on Unmanaged Switches, page 4-7](#)

You can use FACT to install firmware on Cisco Server Fabric Switches running SFS OS, both on managed and unmanaged switches. Likewise, you can install firmware either on a single switch or on multiple switches simultaneously. (This firmware upgrade feature is not available for the OEM switches: Cisco SFS 7012 or Cisco SFS 7024.)



### Note

Because each firmware image is for a specific switch model, if you upgrade firmware on multiple switches simultaneously, all switches must be of the same model for which the firmware file is intended. FACT returns an error message if the firmware image does not match the switch model.

## Installing Firmware in Managed Switches

To install firmware on a managed Cisco Server Fabric Switch, perform the following steps:

- Step 1** Log in to the host as a superuser.
- Step 2** Enter the **install** command to obtain and install the new firmware image. For an SFS OS switch, FACT instructs the switch to download the firmware from an FTP server, which you must set up in advance. The *path* argument must be an FTP URL with the following syntax:  

```
ftp://user:password@host/path/file
```
- Step 3** FACT checks the firmware on each switch that you want to upgrade and prints version numbers to the screen, grouped by version number. FACT then asks for confirmation. Confirm the version by typing **Y**. FACT installs the new firmware on all specified switches.

The following example shows an SFS OS firmware upgrade from build 137 to build 154 on switches 1 through 2. The example shows that while the user attempted to upgrade switch 3, FACT determined that switch 3 was already running the updated build, and FACT disregarded the installation:

```
fact> install switch[1-3] from
ftp://user:passwd@host/path/Topspin90-TopspinOS-2.8.0-build154.img
Checking switch1
Checking switch2
Checking switch3
These switches are running TopspinOS 2.8.0 build 137:
switch1
```

```

switch2
These switches are already running TopspinOS 2.8.0 build 154:
switch3

Install [Y/n]? y
Installing switch1
Installing switch2
Skipping switch3
Waiting for switch1
Waiting for switch2
Waiting for switch3
switch1 rebooted
switch2 rebooted
switch3 rebooted
Rescan fabric to update switch versions

```

- Step 4** Enter the **scan fabric** command so that FACT can learn the current version that the switches are running.
- Step 5** Enter the **show versions** command to ensure that the correct version appears in the output.
- 

## Installing Firmware on Unmanaged Switches

For unmanaged switches, FACT uses the ibspark tool to install firmware. The ibspark tool transmits the firmware through InfiniBand to the switch, so the tool must run on a host that is directly connected to the InfiniBand network. However, ibspark is not required to reside on the same host as FACT. FACT can use SSH to reach the host with ibspark. (For information about obtaining the ibspark tool, see the “[Obtaining and Installing ibspark](#)” section.) (For information about configuring FACT to use ibspark, see “[About Master Configuration Files](#)” section.)

To install firmware on an unmanaged Cisco Server Fabric Switch, perform the following steps:

- Step 1** Log in to the host as a superuser.
- Step 2** Enter the **install** command to obtain and install the new firmware image. For an unmanaged switch, the file parameter is a regular firmware image file. A firmware image file ends in .img.
- Step 3** FACT checks the firmware version on each switch that you want to upgrade and prints the version numbers to the screen, grouped by version number. FACT then asks for confirmation. Confirm the version by typing **Y**. FACT installs the new firmware on all specified switches.

The following example shows a firmware upgrade on unmanaged switches 4 through 6:

```

fact> install switch[4-6] from somefile.img
Checking switch4
Checking switch5
Checking switch6
These switches are running version 1.2.3:
switch4
switch5
switch6

Install [Y/n]? y
Installing switch4
Installing switch5
Skipping switch6
Switches are rebooting. Rescan fabric after network stabilizes.
fact>

```

- Step 4** Enter the **scan fabric** command so that FACT can learn the current version that the switches are running.
- Step 5** Enter the **show versions** command to ensure that the correct version appears in the output.
-



## CHAPTER 5

# FACT Command Reference

---

This chapter describes the commands used in FACT and includes the following sections:

- [Using the CLI, page 5-1](#)
- [Command-Line Interface, page 5-5](#)
- [Commands, page 5-10](#)
- [Command Groups, page 5-53](#)

## Using the CLI

This section describes how to use the FACT CLI and includes the following topics:

- [Using FACT Command Arguments, page 5-1](#)
- [Using Command Modes, page 5-4](#)
- [Command-Line Options, page 5-5](#)
- [Environment Variables, page 5-7](#)
- [Editing the CLI, page 5-8](#)
- [Editing the CLI, page 5-8](#)

## Using FACT Command Arguments

An argument is a text expression that denotes one or more objects in the fabric. Arguments are used as arguments to the FACT commands.

This section includes the following topics:

- [Using Singular Arguments, page 5-2](#)
- [Using Plural Arguments, page 5-3](#)

## Using Singular Arguments

This section describes the syntax for each type of argument.

[Table 5-1](#) lists the argument syntax and the corresponding descriptions.

**Table 5-1** *Argument Syntax*

| Syntax              | Argument Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>scan</i>         | Scans are numbered chronologically starting from 1. (1 is the oldest; N is the latest.) The latest scan can be specified as “latest.” The <b>show history</b> command shows the number of each scan.                                                                                                                                                                                                                                                                                                                                                                                                        |
| <i>file</i>         | A file or pathname is used by the FACT host platform.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <i>switch</i>       | A switch can be specified in one of the following ways: <ul style="list-style-type: none"> <li>hostname or IP address of its management port, as specified in the managed-node definition. See <a href="#">“About Managed Node Definitions Files”</a> section.</li> <li>chassis GUID, using the notation 00:11:22:33:44:55:66:77.</li> <li>given name of its chassis GUID (from guid name definition).</li> <li>description of one of its chips, prepended by “chassis,” as shown in the following example:<br/> chassis:00:11:22:33:44:55:66:77 (a node GUID)</li> </ul>                                   |
| <i>host</i>         | A host can be specified in one of the following ways: <ul style="list-style-type: none"> <li>hostname or IP address, as specified in the managed-node definition.</li> <li>given name of its HCA node GUID (from guid name definitions).</li> <li>node GUID of its HCA, prepended by “chassis.”</li> </ul>                                                                                                                                                                                                                                                                                                  |
| <i>chassis</i>      | A chassis is either a host or a switch. Any of the previously listed notations for host or switch may be used.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <i>managed-node</i> | A managed node is either a host or a switch. It must be specified exactly as it appears in the managed-node part of the configuration. See <a href="#">“About Managed Node Definitions Files”</a> section.                                                                                                                                                                                                                                                                                                                                                                                                  |
| <i>chip</i>         | A chip is an InfiniBand switch chip or InfiniBand channel adapter chip. It may be specified in the following ways: <ul style="list-style-type: none"> <li>node GUID.</li> <li>given name from the GUID name definitions.</li> <li>the notation <i>switch/slot:index</i>, where <i>switch</i> is a switch name, as described above, <i>slot</i> is a slot number (decimal integer) in a switch, and <i>index</i> is a decimal number. FACT orders chips in a slot by the node GUID. FACT numbers chips from 1 to n, ordered by their node GUIDs, going in ascending order from lowest to highest.</li> </ul> |

Table 5-1 Argument Syntax (continued)

| Syntax      | Argument Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>port</i> | <p>A port may be specified as a port GUID, relative to a chassis or relative to a chip. As a port GUID, it is specified in eight hexadecimal octets separated by colons, as shown in the following example:</p> <pre>00:02:c9:02:00:22:2a:c8</pre> <p> <b>Note</b> Most switch chips do not have port GUIDs</p> <hr/> <p>Chassis-relative form has the syntax <i>switch/slot/ext-number</i>, where <i>ext-number</i> is the external port number. <i>Switch</i> has any of the previously listed forms, as shown in the following example:</p> <pre>switch01/2/3</pre> <p>(port 3 on slot 2 of chassis switch01)</p> <p> <b>Note</b> Internal chips can be specified in chip-relative syntax only.</p> <hr/> <p>Chip-relative form has the syntax <i>chip/internal-number</i>, where <i>chip</i> is an InfiniBand chip as previously listed, and <i>internal number</i> is the port number on the chip, as shown in the following examples:</p> <pre>switch01/2:3/4</pre> <p>(chip port 4 on chip 3 of the card in slot 2 of chassis switch01)</p> <pre>00:02:c9:02:00:22:2a:c4/4</pre> <p>(chip port 4 on the chip with the given GUID)</p> |
| <i>LID</i>  | A LID is a decimal integer.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <i>MLID</i> | A multicast LID is a decimal integer.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

## Using Plural Arguments

Many of the commands in FACT can accept plural arguments. The plurals can be built in several ways. Several alternatives can be separated by commas, with no spaces, as shown in the following example:

```
switch01,switch02
```

Each alternative may contain one or more ranges. A range is surrounded by square brackets and contains one or more comma-separated alternatives. Each alternative may be a single value or a hyphen-separated range of values, as shown in the following example:

```
rack[A-C,E]-host[01-10]
```

(rackA-host01 through rackA-host10, and similarly for racks B, C, and E, skipping rackD)

Scans, LIDs, and slot, chip, and port numbers are always numeric, so ranges may be specified without the square brackets, as shown in the following example:

```
switch03/1/1-4
```

(ports 1 through 4 on slot 1 of chassis switch03)

A plural scan always describes a single object in a simple form (by not using any alternatives or images), as shown in the following example:

```
switch03
```

FACT uses the following plural arguments:

- *LIDs*
- *chassis*
- *chips*
- *hosts*
- *managed-nodes*
- *multicast-groups*
- *ports*
- *scans*
- *switches*

**Note**

---

Because all keywords can be abbreviated to a unique prefix, plural keywords such as *chips* and *ports* can be abbreviated to *chip* and *port* for the most common case of referring to a single unit. The keyword *chassis* can be either singular or plural.

---

## Using Command Modes

FACT has two modes of operation:

- Interactive command-line processor mode
- Single-use command mode that performs a single function

This section includes the following topics:

- [Using Interactive Mode, page 5-4](#)
- [Using Single-Command Mode, page 5-5](#)

## Using Interactive Mode

As a command-line program, FACT reads commands from its input, parses them, and writes their output to its output. The following sample output shows FACT interactive mode:

```
$ fact
Cisco FACT 1.0
Copyright 2008, Cisco Systems, Inc.
fact> scan fabric
fact> show versions
name version

switch01TopspinOS 2.9.0 releng #147 10/25/...
switch02TopspinOS 2.8.0 releng #154 07/11/...
fact> exit
$
```

## Using Single-Command Mode

In single-command mode, a single command is passed to FACT as the program command arguments. FACT then runs the command, prints the results, and exits.

The following sample output shows the FACT single-command mode:

```
$ fact scan fabric
$ fact show versions
node version

switch01TopspinOS 2.9.0 releng #147 10/25/...
switch02TopspinOS 2.8.0 releng #154 07/11/...
$
```

As a variant of single-command mode, the first word of the command, usually the verb, can be joined to the FACT name with a hyphen. The following sample output shows the FACT single-command mode:

```
$ fact-scan fabric
$ fact-show versions
```

To use shell completion to see available commands, join the FACT name with a hyphen, and enter the tab key:

```
$ fact-<TAB>
```

## Command-Line Interface

FACT looks for several environmental variables and accepts several command-line options.

This section describes the FACT command-line interface and includes the following topics:

- [Command-Line Options, page 5-5](#)
- [Environment Variables, page 5-7](#)

## Command-Line Options

[Table 5-2](#) lists and describes the command-line options.

**Table 5-2** Command-Line Options

| Option                                 | Description                                                                                                                          |
|----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| <b>-c</b> <i>file</i>                  | Reads primary configuration from the specified file.                                                                                 |
| <b>--config-file</b> = <i>file</i>     |                                                                                                                                      |
| <b>--color=always</b>                  | If tracing, specifies that the transcript is colorized, showing FACT requests in one color and the switch responses in another.      |
| <b>--color=auto</b>                    | If tracing, specifies that the transcript is colorized only if the FACT standard output is a terminal. This behavior is the default. |
| <b>--color=never</b>                   | If tracing, specifies that the transcript is not colorized.                                                                          |
| <b>--config-option</b> = <i>option</i> | Adds or overrides a configuration option.                                                                                            |

Table 5-2 Command-Line Options (continued)

| Option                                                       | Description                                                                                                                                                                                                                                                                                                                                                          |
|--------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>--credentials-from</b><br><i>=file</i>                    | Adds credential information from the file.<br>See the “ <a href="#">About Credentials Files</a> ” section.                                                                                                                                                                                                                                                           |
| <b>--guid-name=guid</b><br><i>=name</i>                      | Adds a GUID name definition.<br>See the “ <a href="#">About GUID Name Definitions</a> ” section.                                                                                                                                                                                                                                                                     |
| <b>--guid-names-from=</b> <i>file</i>                        | Adds GUID name definitions from the specified file.<br>See the “ <a href="#">About GUID Name Definitions</a> ” section.                                                                                                                                                                                                                                              |
| <b>-h</b><br><b>--help</b>                                   | Prints a message describing messages.                                                                                                                                                                                                                                                                                                                                |
| <b>--help-commands</b>                                       | Lists FACT commands.                                                                                                                                                                                                                                                                                                                                                 |
| <b>--managed-nodes-</b><br><i>=type: pattern</i>             | Adds a set of managed-nodes to those about which FACT is aware.<br>See the “ <a href="#">About Managed Node Definitions Files</a> ” section.                                                                                                                                                                                                                         |
| <b>--managed-nodes-</b><br><b>from=</b> <i>file</i>          | Adds managed-nodes from the file.<br>See the “ <a href="#">About Managed Node Definitions Files</a> ” section.                                                                                                                                                                                                                                                       |
| <b>-s</b><br><b>--sequential</b>                             | Initiates sequential scan sessions to one managed node at a time. This option is helpful with the <b>--trace</b> command because output displays slower and more predictably.<br><br>By default, FACT connects up to 50 managed nodes at a time during scans and installations. If <b>sequential</b> is specified, then FACT connects to one managed node at a time. |
| <b>--subnet-manager</b><br><i>=domain name or IP address</i> | Specifies the InfiniBand Subnet Manager host that FACT should use.                                                                                                                                                                                                                                                                                                   |
| <b>--trace</b> and<br><b>--sequential</b>                    | Runs sequential sessions to managed nodes and prints the session transcript.                                                                                                                                                                                                                                                                                         |
| <b>-t</b><br><b>--trace</b>                                  | By default, if <b>--sequential</b> is not also specified, then FACT will print the transcript of each session when that session is closed. If <b>--sequential</b> is also specified, FACT will print the transcript of each session as it occurs.                                                                                                                    |
| <b>-v</b><br><b>--version</b>                                | Prints FACT version information and exits.                                                                                                                                                                                                                                                                                                                           |

## Environment Variables

FACT uses environment variables to control the FACT configuration.

[Table 5-3](#) lists and describes the FACT environment variables.

**Table 5-3** *Environment Variables*

| Variable                | Description                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FACT_CONF               | Name of the master configuration file.<br>See the <a href="#">“About Master Configuration Files”</a> section.                                                                                                                                                                                                                                                                                                      |
| FACT_CREDENTIAL_FILES   | List of files that contain credentials, separated by colons, as shown in the following example:<br><br>FACT_CREDENTIAL_FILES=/etc/fact/cred1:/etc/fact/cred2<br>states that the files /etc/fact/cred1 and /etc/fact/cred2 contain GUID name definitions<br><br>See the <a href="#">“About Credentials Files”</a> section.                                                                                          |
| FACT_GUID_NAMES         | List of GUID name definitions, separated by semicolons. Each definition has the form “GUID=name,” as shown in the following example:<br><br>FACT_GUID_NAMES="00:1b:0d:0b:00:00:ca:a6=switchA;00:1b:0d:0b:00:00:ca:aa=switchB"<br>defines switchA as the name for 00:1b:0d:0b:00:00:ca:a6 and switchB as the name for 00:1b:0d:0b:00:00:ca:aa<br><br>See the <a href="#">“About GUID Name Definitions”</a> section. |
| FACT_GUID_NAME_FILES    | List of files containing GUID name definitions, separated by colons, as shown in the following example:<br><br>FACT_GUID_NAME_FILES=/tmp/guidnames1:/tmp/guidnames2<br>states that the files /tmp/guidnames1 and /tmp/guidnames2 contain GUID name definitions<br><br>See the <a href="#">“About GUID Name Definitions”</a> section.                                                                               |
| FACT_MANAGED_NODES      | In-line list of managed node definitions, separated by colons, as shown in the following example:<br><br>FACT_MANAGED_NODES="host=ibmg-r1-evil:SFSOS<br>switch=172.29.232.45"<br>defines a host named ibmg-r1-evil and an SFS OS switch named 172.29.232.45<br><br>See the <a href="#">“About Managed Node Definitions Files”</a> section.                                                                         |
| FACT_MANAGED_NODE_FILES | List of files containing managed node definitions, separated by colons, as shown in the following example:<br><br>FACT_MANAGED_NODE_FILES=/tmp/nodefile1:/tmp/nodefile2<br>states that the files /tmp/nodefile1 and /tmp/nodefile2 contain managed node definitions<br><br>See the <a href="#">“About Managed Node Definitions Files”</a> section.                                                                 |

## Correcting Commands

The system response to command line-errors is different when you use the question mark (?) to obtain help for a command. In this case, the system repeats your input following the subsequent prompt for ease of editing, as shown in the following example:

```
fact> show interface ?
 ^
% Error: Unrecognized command
fact> show interface
```

## Editing the CLI

Command-line editing lets you modify a command line command that you have just entered or a command line that you entered previously in the CLI session. The CLI supports a variety of ways to move about and edit the currently displayed command line. [Table 5-4](#) lists and describes these options.

**Table 5-4** Key Stroke Shortcuts

| Key Strokes | Description                                                                         |
|-------------|-------------------------------------------------------------------------------------|
| Ctrl-A      | Moves the cursor to the beginning of the line.                                      |
| Ctrl-B      | Moves the cursor left (backward) one character.                                     |
| Ctrl-D      | Deletes the character at the cursor. Exits FACT if the current input line is empty. |
| Ctrl-E      | Moves the cursor to the end of the line.                                            |
| Ctrl-F      | Moves the cursor to the right (forward) one character.                              |
| Ctrl-G      | Erases the current input line.                                                      |
| Ctrl-H      | Deletes the character to the left of the cursor, analogous to Backspace key.        |
| Ctrl-I      | Completes the current word, analogous to Tab key.                                   |
| Ctrl-J      | Accepts the current line.                                                           |
| Ctrl-K      | Deletes text from cursor to the end of the line.                                    |
| Ctrl-L      | Refreshes the input line.                                                           |
| Ctrl-M      | Accepts the current line, analogous to Return key.                                  |
| Ctrl-N      | Proceeds to the next line in the command history.                                   |
| Ctrl-P      | Moves to the previous line in the command history.                                  |
| Ctrl-T      | Transposes the current and previous characters.                                     |
| Ctrl-U      | Deletes all text to the left of the cursor, back to the beginning of the line.      |
| Ctrl-W      | Deletes the word to the left of the cursor.                                         |
| Ctrl-Z      | Suspends FACT.                                                                      |
| Ctrl-?      | Deletes the character to the left of the cursor, analogous to Delete key.           |
| ?           | Completes the current word.                                                         |
| Esc-[A      | Moves to the previous line in the command history.                                  |
| Esc-[B      | Proceeds to the next line in the command history.                                   |
| Esc-[C      | Moves the cursor to the right (forward) one character.                              |

**Table 5-4** *Key Stroke Shortcuts (continued)*

| Key Strokes | Description                                                                |
|-------------|----------------------------------------------------------------------------|
| Esc-[D      | Moves the cursor to the left (backward) one character.                     |
| Esc-B       | Moves the cursor to the left (backward) one word.                          |
| Esc-C       | Converts characters, from the cursor to the end of the word, to uppercase. |
| Esc-D       | Deletes characters from the cursor through end of the word.                |
| Esc-F       | Moves the cursor right (forward) one word.                                 |
| Esc-L       | Converts characters, from the cursor to the end of the word, to lowercase. |
| Esc-U       | Converts characters, from the cursor to the end of the word, to uppercase. |
| Esc-Ctrl-H  | Deletes the word to the left of the cursor.                                |
| Esc-Delete  | Deletes the character to the left of the cursor.                           |
| down-arrow  | Proceeds to the next line in the command history.                          |
| up-arrow    | Moves to the previous line in the command history.                         |
| left-arrow  | Moves the cursor to the left (backward) one character.                     |
| right-arrow | Moves the cursor to the right (forward) one character.                     |

# Commands

FACT software uses the commands in this section, which are listed in alphabetical order. If you prefer a structured organization with commands grouped by function, see the [“Command Groups”](#) section.

FACT uses the following commands:

- **annotate**
- **delete scans**
- **disable ports**
- **enable ports**
- **exit**
- **export repository**
- **export scan**
- **help**
- **import repository**
- **import scan**
- **install**
- **ports speed**
- **reset port-counters**
- **scan fabric**
- **scan fabric all**
- **scan tech-support**
- **select**
- **show changes scan**
- **show chassis**
- **show chips**
- **show counters monitored**
- **show counters raw**
- **show counters threshold**
- **show destinations**
- **show egress ports**
- **show history**
- **show isolated managed-nodes**
- **show multicast destinations**
- **show multicast egress ports**
- **show multicast groups**
- **show multicast routes**
- **show ports**
- **show routes**

- [show subnet-managers](#)
- [show versions](#)

# annotate

To annotate one or more syslog files and print them to the screen with annotations attached, use the **annotate** command.

**annotate** [pass-through | filtered] [follow] [*files*]

| Syntax Description |            |                                                                                                                                  |
|--------------------|------------|----------------------------------------------------------------------------------------------------------------------------------|
| pass-through       | (Optional) | Indicates that all messages are printed to standard output.                                                                      |
| filtered           | (Optional) | Indicates that only those messages that are deemed important by FACT are printed.                                                |
| follow             | (Optional) | Specifies that FACT continuously watches for more records to be appended to the files and analyzes those records as they arrive. |
| <i>files</i>       | (Optional) | Specifies the files to annotate (one file or multiple files).                                                                    |

**Command Default** The default is **pass-through**.

**Usage Guidelines** This command uses the current scan to perform annotations. If you specify no files, FACT looks in the FACT main configuration file and annotates only the files that are in log-file parameters.

**Examples** The following example shows how FACT annotates a short log file:

```
fact> annotate mylogfile
```

```
Mar 21 20:14:28 herosm1 ib_sm[30456]: %IB-6-INFO: Generate SM IN_SERVICE trap for
GID=fe:80:00:00:00:00:00:00:18:8b:90:97:fe:13:3a (hero0404/1/1, hero0404/1:1/1)
(hero0404/1/1, hero0404/1:1/1, neighbor = 00:1b:0d:0b:00:00:8e:63 (leaf04b)/1/4)
```

**Related Commands** [ports speed](#)

# delete scans

To delete one or more scans from the repository, use the **delete scans** command.

**delete scans** [*scans*]

|                           |              |                                                                                                    |
|---------------------------|--------------|----------------------------------------------------------------------------------------------------|
| <b>Syntax Description</b> | <i>scans</i> | (Optional) Deletes the specified scan, the specified range of scans, or the latest scan performed. |
|---------------------------|--------------|----------------------------------------------------------------------------------------------------|

**Command Default** No default behavior or values.

**Usage Guidelines** Range numbers correspond to the numbers in the [show history](#) command.

**Examples** The following example shows deleting scan 2 and scan 3 from the repository:

```
fact> show history
```

| index | current | type         | date                    | subnet-manager |
|-------|---------|--------------|-------------------------|----------------|
| 1     |         | fabric       | 2008-04-17 10:44:02 PDT | host01         |
| 2     |         | fabric       | 2008-04-17 10:44:08 PDT | host01         |
| 3     |         | fabric       | 2008-04-17 10:44:14 PDT | host01         |
| 4     | Y       | tech-support | 2008-04-17 10:44:47 PDT | host01         |

```
fact> delete scans 2-3
Proceed with scan delete [Y/n]? yes
```

```
fact> show history
```

| index | current | type         | date                    | subnet-manager |
|-------|---------|--------------|-------------------------|----------------|
| 1     |         | fabric       | 2008-04-17 10:44:02 PDT | host01         |
| 2     | Y       | tech-support | 2008-04-17 10:44:47 PDT | host01         |

**Related Commands** [show history](#)  
[select](#)

# disable ports

To disable a single port or multiple ports in the network, use the **disable ports** command.

**disable ports** { *chassis* | *chips* | *ports* } [internal | external]

| Syntax Description |                |                                                                |
|--------------------|----------------|----------------------------------------------------------------|
|                    | <i>chassis</i> | Specifies the chassis in which to disable the specified ports. |
|                    | <i>chips</i>   | Specifies the chips in which to disable the specified ports.   |
|                    | <i>ports</i>   | Specifies the ports to disable.                                |
|                    | internal       | (Optional) Disables the specified internal ports only.         |
|                    | external       | (Optional) Disables the specified external ports only.         |

**Command Default** No default behavior or values.

**Usage Guidelines** Only physical switch ports can be disabled. HCA ports cannot be disabled.  
For isolated managed nodes, only external ports can be configured, and only for SFS OS switches and OEM switches, not for hosts.



**Note**

For unmanaged switches, if disabling the port results in the device being isolated from the rest of the IB fabric, the only way to reenabale that port is by resetting the unmanaged switch to restore its original configuration.

**Examples** The following example shows disabling port 4, slot 1 in switchA:

```
fact> disable ports switchA/1/4
```

```
The following port(s) will be configured:
switchA/1/4
```

```
Proceed with port configuration [Y/n]? y
Disabling port switchA/1/4
```

**Related Commands**

- [show ports](#)
- [ports speed](#)
- [enable ports](#)

# enable ports

To enable a single port or multiple ports in the network, use the **enable** command.

**enable ports** { *chassis* | *chips* | *ports* } [internal | external]

| Syntax Description |            |                                                             |
|--------------------|------------|-------------------------------------------------------------|
| <i>chassis</i>     |            | Specifies the chassis in which to enable the ports.         |
| <i>chips</i>       |            | Specifies the chips in which to enable the specified ports. |
| <i>ports</i>       |            | Specifies the ports to enable.                              |
| internal           | (Optional) | Enables the specified internal ports only.                  |
| external           | (Optional) | Enables the specified external ports only.                  |

**Command Default** No default behavior or values.

**Usage Guidelines** Only physical switch ports can be enabled. HCA ports cannot be enabled.  
For isolated managed nodes, only external ports can be configured, and only for SFS OS switches and OEM switches, not for hosts.

**Examples** The following example shows enabling port 4, slot1 in switchA:

```
fact> enable ports switchA/1/4
```

```
The following port(s) will be configured:
switchA/1/4
```

```
Proceed with port configuration [Y/n]? y
Enabling port switchA/1/4
```

**Related Commands** [disable ports](#)

[ports speed](#)

[show ports](#)

# exit

To discontinue the current session with FACT, use the **exit** command.

**exit**

---

**Syntax Description** This command has no arguments or keywords.

---

**Command Default** No default behavior or values.

# export repository

To make a copy of the specified repository and save a copy of FACT repository in a file, use the **export repository** command.

```
export repository to {file}
```

---

**Syntax Description**

|             |                                                         |
|-------------|---------------------------------------------------------|
| <i>to</i>   | Identifies the destination set.                         |
| <i>file</i> | Exports the current repository into the specified file. |

---

---

**Command Default**

No default behavior or values.

---

**Examples**

The following example shows exporting the repository to a file named “myrepo”:

```
fact> export repository to myrepo
Overwrite existing file myrepo {Y/n} yes
```

---

**Related Commands**

[export scan](#)  
[import repository](#)  
[import scan](#)

# export scan

To export the current scan to a file, use the **export scan** command.

**export scan** to *{file}*

## Syntax Description

|             |                                                 |
|-------------|-------------------------------------------------|
| <i>to</i>   | Identifies the destination set.                 |
| <i>file</i> | Exports the current scan to the specified file. |

## Defaults

No default behavior or values.

## Examples

The following example shows exporting the current scan to a file named “myscan”:

```
fact> export scan to myscan
Overwrite existing file myscan [Y/no?] yes
```

## Related Commands

[export repository](#)  
[import repository](#)  
[import scan](#)

# help

To list the complete FACT grammar or to show information about specific commands, use the **help** command.

```
help [help | commands | grammar] [{help command}]
```

## Syntax Description

|                |                                                                   |
|----------------|-------------------------------------------------------------------|
| help           | (Optional) Shows general information about obtaining help.        |
| commands       | (Optional) Displays a one-line summary about each command.        |
| grammar        | (Optional) Displays the syntax of all commands.                   |
| <i>command</i> | (Optional) Displays a short description of the specified command. |

## Command Default

If you enter the **help** command, where *command* is ambiguous, then FACT displays a set of commands that match what you entered.

## Examples

The following example shows entering the **help** command, where *command* is ambiguous. FACT displays a set of commands that match what you entered:

```
fact> help scan
Scanning
 scan fabric - collect information about the network.
 scan fabric all - collect lots of information about the network.
 scan tech-support - collect diagnostic information about the network.
fact> help show ch
Subnet-Level Display
 show chassis - show information about several chassis.
 show chips - show information about chips (InfiniBand nodes).
```

# import repository

To replace the existing repository with the one that is currently in the file, use the **import repository** command.

**import repository** from *{file}*

## Syntax Description

|             |                                                         |
|-------------|---------------------------------------------------------|
| from        | Identifies the source set.                              |
| <i>file</i> | Imports the current repository into the specified file. |

## Defaults

No default behavior or values.

## Usage Guidelines

When you import a new repository with this command, you overwrite the existing repository and lose all data from the previous repository.

## Examples

The following example shows replacing the existing repository with the repository previously exported in the file named “myrepo”:

```
fact> import repository from myrepo
Overwrite existing repository [Y/n]? yes
```

## Related Commands

[export repository](#)  
[export scan](#)  
[import scan](#)

# import scan

To add the scan to the current repository, use the **import scan** command.

**import scan** from *{file}*

---

**Syntax Description**

|             |                                                     |
|-------------|-----------------------------------------------------|
| from        | Identifies the source set.                          |
| <i>file</i> | Indicates the file from which the scan is imported. |

---

---

**Command Default**

No default behavior or values.

---

**Examples**

The following example shows importing the scan previously exported to the file “mysecan,” making it the current scan:

```
fact> import scan from myscan
Overwrite existing scan in repository [Y/n]? yes
```

---

**Related Commands**

[export repository](#)  
[export scan](#)  
[import repository](#)

# install

FACT can install new firmware on an SFS OS switch or on an unmanaged switch. To install a new firmware image into an individual switch or a set of switches, use the **install** command. (For more information about installing firmware using FACT, see the “[Installing Firmware with FACT](#)” section.)

```
install {switches} from {path}
```

## Syntax Description

|                 |                                                                                                                                                                                                                                                                          |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>switches</i> | Identifies the switches on which to install the firmware.                                                                                                                                                                                                                |
| from            | Identifies the source set.                                                                                                                                                                                                                                               |
| <i>path</i>     | Identifies the firmware file to install.<br><br>For an SFS OS switch, <i>path</i> must be a path with the following format:<br><b>ftp://user: password@host/path/file/build.img</b><br><br>For an unmanaged switch, <i>file</i> is a user-specified firmware image file. |

## Command Default

No default behavior or values.

## Usage Guidelines

FACT cannot install firmware into OEM switches.



### Note

Because each firmware image is designed for a specific switch model, if you update a group of switches at one time, all switches must be of the same model.

For unmanaged switches, FACT copies the new firmware image file to the ibspark host and uses ibspark to install it. For information about obtaining ibspark, see the “[Obtaining and Installing ibspark](#)” section.

The **install** command runs for three to four minutes and provides minimal feedback while running. To monitor progress, use the **--trace** command-line argument if you are installing firmware on multiple switches. If you are installing on a single switch, use the **--trace** and **--sequential** arguments. For more information about command-line options, see [Table 5-2](#), “Command-Line Options.”



### Note

When you install firmware on an SFS OS switch that is configured to be the master Subnet Manager, wait for at least one minute after the installation for the Subnet Manager to take control of the subnet before rescanning the fabric, or the scan fails and no master Subnet Manager is found.

## Examples

The following example shows how to install firmware on an SFS OS switch in switchA:

```
fact> install switchB from
ftp://username:password@ftphost/pub//Topspin120d-TopspinOS-2.9.0-build163.img
Checking switchB
These switches are running SFS-7000D version 2.11.0 build 91:
switchB

Install [Y/n]? yes
Installing switchB
Waiting for switchB to reboot
```

```
Switch switchB rebooted
Rescan fabric to update switch versions.
fact> scan fabric
Scanning host01
Scanning switchA
Scanning switchB
Scanning Master Subnet Manager at host01s
```

The following example shows installing firmware on an unmanaged switch.

```
fact> install switchC from myfirmware.img
Checking switchC
These switches are already running version 1.0.0:
 switchC

Install anyway [y/N]? yes
Installing switchC
Switches are rebooting. Rescan fabric after network stabilizes.
fact> scan fabric
Scanning localhost
Scanning Master Subnet Manager on localhost
```

**Related Commands**    [show versions](#)

# ports speed

To set the width and speed of a single port or multiple ports in the network, use the **ports speed** command.

```
ports speed { chassis | chips | ports } [external | internal] { 1x-sdr | 1x-ddr | 4x-sdr | 4x-ddr | 12x-sdr
| 12x-ddr | auto }
```

## Syntax Description

|                |                                                                                                                         |
|----------------|-------------------------------------------------------------------------------------------------------------------------|
| <i>chassis</i> | Specifies the chassis (one or more) for which you want to set the port speed on all ports.                              |
| <i>chips</i>   | Specifies the switch chip or channel adapter chips (one or more) for which you want to set the port speed on all ports. |
| <i>ports</i>   | Specifies the ports for which you want to set the speed.                                                                |
| external       | (Optional) Restricts the command to operate on external ports only.                                                     |
| internal       | (Optional) Restricts the command to operate on internal ports only.                                                     |
| 1x-sdr         | Specifies that the port is set to 1x Single Data Rate.                                                                  |
| 1x-ddr         | Specifies that the port is set to 1x Double Data Rate.                                                                  |
| 4x-sdr         | Specifies that the port is set to 4x Single Data Rate.                                                                  |
| 4x-ddr         | Specifies that the port is set to 4x Double Data Rate.                                                                  |
| 12x-sdr        | Specifies that the port is set to 12x Single Data Rate.                                                                 |
| 12x-ddr        | Specifies that the port is set to 12x Double Data Rate.                                                                 |
| auto           | Specifies that the port automatically negotiates the optimal width and speed.                                           |

## Command Default

No default behavior or values.

## Usage Guidelines

FACT first determines whether or not the port can support the requested width and speed combination. If the port cannot, FACT returns an error message.

For isolated managed nodes, only external ports can be configured, and only for SFS OS switches and OEM switches, not for hosts.

## Examples

The following example shows setting the switch to 4x and the speed to Single Data Rate on switchA, slot1, port 3:

```
fact> port speed switchA/1/3 4x-sdr
```

```
The following port(s) will be configured:
switchA/1/3
```

```
Proceed with port configuration [Y/n]? y
Configuring speed on port switchA/1/3
```

## Related Commands

[enable ports](#)

[disable ports](#)

**show ports**

# reset port-counters

To tell the master Subnet Manager to reset all port counters to zero, use the **reset port-counters** command.

## **reset port-counters**

---

**Syntax Description** This command has no arguments or keywords.

---

**Command Default** No default behavior or values.

---

**Examples** The following example shows resetting all port counters on all switches:

```
fact> reset port-counters
Resetting port counters via Subnet Manager host01
```

---

**Related Commands**

- [scan fabric](#)
- [scan fabric all](#)
- [show counters monitored](#)
- [show counters raw](#)
- [show counters threshold](#)

# scan fabric

To scan the static connectivity of the subnet and collect information from and about the Subnet Managers, use the **scan fabric** command.

```
scan fabric [routes] [raw-counters] [monitored-counters] [threshold-counters] [subnet-manager |
sm managed-node]
```

| Syntax Description  |                                                                             |
|---------------------|-----------------------------------------------------------------------------|
| routes              | (Optional) Collects routing information from the master Subnet Manager.     |
| raw-counters        | (Optional) Collects raw port counters from the master Subnet Manager.       |
| monitored counters  | (Optional) Collects monitored port counters from the master Subnet Manager. |
| threshold-counters  | (Optional) Collects threshold-exceeded port counters.                       |
| subnet-manager      | (Optional) Overrides the discovered master Subnet Manager.                  |
| sm                  | (Optional) Overrides the discovered master Subnet Manager.                  |
| <i>managed-node</i> | (Optional) Scans the configured managed-nodes.                              |

**Command Default** No default behavior or values.

**Usage Guidelines** The scan will fail if the specified managed-node is not a master Subnet Manager.

**Examples** The following example shows scanning the fabric and also collecting route information and raw port counter information:

```
fact> scan fabric routes raw-counters
Scanning host01
Scanning switchA
Scanning switchB
Scanning Master Subnet Manager at host 01
```

**Related Commands**

- [scan fabric all](#)
- [scan tech-support](#)
- [select](#)

# scan fabric all

To perform a fabric scan, which scans the static connectivity of the subnet, and to collect routing information, raw port counter information, monitored port counter information, and threshold-exceeded port counter information from the Subnet Manager, use the **scan fabric all** command.

```
scan fabric all [subnet-manager | sm managed-node]
```

| Syntax Description |                     |                                                                                           |
|--------------------|---------------------|-------------------------------------------------------------------------------------------|
|                    | subnet-manager      | (Optional) Overrides the discovered master Subnet Manager.                                |
|                    | sm                  | (Optional) Overrides the discovered master Subnet Manager.                                |
|                    | <i>managed-node</i> | (Optional) Scans the configured managed-nodes where the master Subnet Manager is running. |

**Command Default** No default behavior or values.

**Usage Guidelines** The scan will fail if the specified managed-node is not a master Subnet Manager.

**Examples** The following example shows a full fabric scan:

```
fact> scan fabric all
Scanning host01
Scanning switchA
Scanning switchB
Scanning Master Subnet Manager at host 01
```

**Related Commands**

- [scan fabric](#)
- [scan tech-support](#)
- [select](#)

# scan tech-support

To collect technical support information of all managed-nodes in the subnet and save transcripts that can be forwarded to Cisco TAC or engineering, use the **scan tech-support** command.

## **scan tech-support**

**Syntax Description** This command has no arguments or keywords.

**Command Default** No default behavior or values.

**Examples** The following example shows collecting a tech-support scan from all managed nodes:

```
fact> scan tech support
Scanning host01
Scanning switchA
Scanning switchB
Scanning Master Subnet Manager at host 01
```

**Related Commands**

- [export repository](#)
- [export scan](#)
- [scan fabric](#)
- [select](#)

# select

To select the current scan, use the **select** command.

```
select {scan}
```

|                           |             |                                                                                    |
|---------------------------|-------------|------------------------------------------------------------------------------------|
| <b>Syntax Description</b> | <i>scan</i> | Identifies the index number of the scan in the scan history that you want to view. |
|---------------------------|-------------|------------------------------------------------------------------------------------|

**Command Default** No default behavior or values.

**Usage Guidelines** In the command output, the lowest index number in the “show index” column shows the first scan performed, and the highest number is the latest scan performed.

**Examples** The following example shows that before entering the **select** command, scan 2 is current. Afterwards, scan 1 is current:

```
fact> show history
```

| index | current | type         | date                    | subnet-manager |
|-------|---------|--------------|-------------------------|----------------|
| 1     |         | fabric       | 2008-04-17 10:44:02 PDT | host01         |
| 2     | Y       | tech-support | 2008-04-17 10:44:47 PDT | host01         |

```
fact> select 1
fact> show history
```

| index | current | type         | date                    | subnet-manager |
|-------|---------|--------------|-------------------------|----------------|
| 1     | Y       | fabric       | 2008-04-17 10:44:02 PDT | host01         |
| 2     |         | tech-support | 2008-04-17 10:44:47 PDT | host01         |

**Related Commands** [show history](#)

# show changes scan

To compare the current scan with any specified scan in the repository, use the **show changes scan** command.

**show changes scan** *scan* [*chassis* | *chips* | *ports* | subnet managers | sm]

| Syntax Description   |  |                                                                             |
|----------------------|--|-----------------------------------------------------------------------------|
| <i>scan</i>          |  | Specifies the number of a particular scan to compare with the current scan. |
| <i>chassis</i>       |  | (Optional) Shows changes for only the specified chassis.                    |
| <i>chips</i>         |  | (Optional) Shows changes for only the specified chips.                      |
| <i>ports</i>         |  | (Optional) Shows changes for only the specified ports.                      |
| subnet managers   sm |  | (Optional) Reports changes to the subnet managers only.                     |

**Command Default** No default behavior or values.

## Examples

The following is sample output from the **show changes scan** command, and it shows the changes between scan 31 and the current scan:

```
fact> show changes scan 31
Old scan: 2007-10-23 05:46:14 PDT
New scan: 2007-11-01 12:02:21 PDT

Chassis:
No changes.

Chips:
Removed switch04/4:1
Removed switch04/4:2
Removed switch04/4:3
Added switch06/3:1
Added switch06/3:2
Added switch06/3:3

Neighbors:
Removed switch02/1/1 neighbor switch04/4/1
Added switch02/1/1 neighbor switch06/3/1

Subnet Managers:
Removed master host08/1/1
Removed standby host03/1/2
Added master host03/1/2
Added standby host12/1/1
```

## Related Commands

[show chassis](#)  
[show chips](#)  
[show history](#)  
[show ports](#)  
[show subnet-managers](#)

# show chassis

To show the chassis in the current scan, use the **show chassis** command.

**show chassis** [*chassis*] [summary | guides | detail]

| Syntax Description |            |                                                        |
|--------------------|------------|--------------------------------------------------------|
| <i>chassis</i>     | (Optional) | Selects which chassis to show (one or more).           |
| summary            | (Optional) | Shows a one-line summary of each chassis.              |
| guides             | (Optional) | Shows a one-line summary and the GUID of each chassis. |
| detail             | (Optional) | Shows detailed information about each chassis.         |

**Command Default** The default view is the summary view, which shows all chassis on the subnet.

**Usage Guidelines** If there is no current scan or if the current scan is incomplete, using the **show chassis** command automatically performs a fabric scan.

**Examples** The following is sample output from the **show chassis** command, and it shows all chassis in the current scan:

```
fact> show chassis

chassis-name model

host01 Host
switchA SFS-3001
switchB SFS-7000
```

**Related Commands**

- [scan fabric](#)
- [show chips](#)
- [show multicast groups](#)
- [show ports](#)

# show chips

To show information about switch chips in the current scan, use the **show chips** command.

**show chips** [*chassis* | *chips*] [summary | guides | detail | full]

| Syntax Description |            |                                                                 |
|--------------------|------------|-----------------------------------------------------------------|
| <i>chassis</i>     | (Optional) | Shows all chips in the specified chassis (one or more chassis). |
| <i>chips</i>       | (Optional) | Specifies which chip to view.                                   |
| summary            | (Optional) | Shows a one-line summary of each chip.                          |
| guides             | (Optional) | Shows a one-line summary and the node GUID of each chassis.     |
| detail             | (Optional) | Shows the most important details about each chip.               |
| full               | (Optional) | Shows full details about each chip.                             |

**Command Default** The default view is the summary view, which shows all chips in the current scan.

**Usage Guidelines** If there is no current scan or the current scan is incomplete, using the **show chips** command automatically performs a fabric scan.

**Examples** The following is sample output from the **show chips** command, and it shows all chips in the current scan:

```
fact> show chips
```

```

node-name type slot description

host01/1:1 CA 1 kbob-host.cisco.com HCA-1 (Topspin DDR-HCAe MF)
switchA/1:1 switch 1 Topspin Switch - U1
switchA/1:2 switch 1 Topspin Switch - U2
switchA/1:3 switch 1 Topspin Switch - U3
switchB/1:1 switch 1 Cisco Switch SFS7000D

```

**Related Commands**

- [scan fabric](#)
- [show chassis](#)
- [show multicast groups](#)
- [show ports](#)

# show counters monitored

To display the monitored port counters, use the **show counters monitored** command.

**show counters monitored** [chassis / chips / ports] [error]

| Syntax Description |                                                                                                                                                                                     |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| chassis            | (Optional) Displays only the counters for the ports in the specified chassis.                                                                                                       |
| chips              | (Optional) Displays only the counters for the ports in the specified chips.                                                                                                         |
| ports              | (Optional) Displays only the counters for the specified ports.                                                                                                                      |
| error              | (Optional) Displays only those counters that are both error counters and non-zero counters. Some counters, such as xmit-pkts, count non-error events. These counters are not shown. |

**Defaults** No default behavior or values.

**Usage Guidelines** If the current scan does not include monitored port counters, then this command fails. FACT does not display information for ports that have zero error counters only.

**Examples** The following is sample output from the **show counters monitored** command, and it shows the monitored counters that have a non-zero error count:

```
fact> show counters monitored error

port counter value

port: switchB/1/3 symbol_errors 65535
ngbr: switchA/1/3

port: switchB/1/6 symbol_errors 65535
ngbr:

```

**Related Commands**

- [scan fabric](#)
- [scan fabric all](#)
- [show counters raw](#)
- [show counters threshold](#)
- [reset port-counters](#)

# show counters raw

To display the raw port counters, use the **show counters raw** command.

**show counters raw** [chassis / chips / ports] [error]

| Syntax Description |                                                                                                                                                                                     |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| chassis            | (Optional) Displays only the counters for the ports in the specified chassis.                                                                                                       |
| chips              | (Optional) Displays only the counters for the ports in the specified chips.                                                                                                         |
| ports              | (Optional) Displays only those counters for the specified ports.                                                                                                                    |
| error              | (Optional) Displays only those counters that are both error counters and non-zero counters. Some counters, such as xmit-pkts, count non-error events. These counters are not shown. |

**Defaults** No default behavior or values.

**Usage Guidelines** If the current scan does not include raw port counters, then the **show counters raw** command fails. FACT does not display information for ports that have zero error counters only.

**Examples** The following is sample output from the **show counters raw** command, and it shows all non-zero raw counters in the subnet. Errors occurred on host01/1/1, switchB/1/3, switchB/1/6, and switchB/1/7. Three of the four ports have neighbors (nbr), and switchB/1/6 does not have a neighbor. The example also lists the names and values of the non-zero counters:

```
fact> show counters raw error

port counter value

port: host01/1/1 xmit_discards 3
nbr: switchB/1/7 v115_dropped 7

port: switchB/1/3 symbol_errors 65535
nbr: switchA/1/3 link_downs 3
 xmit_discards 13
 xmit_constraint_errors 13

port: switchB/1/6 symbol_errors 65535
nbr: link_downs 1
 xmit_discards 50
 xmit_constraint_errors 50

port: switchB/1/7 rcv_switch_relay_errors 3
nbr: host01/1/1

```

**Related Commands**

- [reset port-counters](#)
- [scan fabric](#)
- [scan fabric all](#)

[show counters threshold](#)

[show counters monitored](#)

# show counters threshold

To display port counters that have exceeded their error thresholds, use the **show counters threshold** command.

**show counters threshold** [chassis | chips | ports] [errors]]

| Syntax Description |                                                                                                                                                                                     |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| chassis            | (Optional) Displays only the counters for the ports in the specified chassis.                                                                                                       |
| chips              | (Optional) Displays only the counters for the ports in the specified chips.                                                                                                         |
| ports              | (Optional) Displays the counters for the specified ports.                                                                                                                           |
| error              | (Optional) Displays only those counters that are both error counters and non-zero counters. Some counters, such as xmit-pkts, count non-error events. These counters are not shown. |

**Defaults** No default behavior or values.

**Usage Guidelines** If the current scan does not include threshold port counters, then this command fails.

**Examples** The following is sample output from the **show counters threshold** command, and it shows all counters in the subnet that have exceeded their own thresholds:

```
fact> show counters threshold

port counter value

port: switchB/1/3 symbol_errors 65535
ngbr: switchA/1/3

port: switchB/1/6 symbol_errors 65535
ngbr:

```

**Related Commands**

- [reset port-counters](#)
- [scan fabric](#)
- [scan fabric all](#)
- [show counters monitored](#)
- [show counters raw](#)

# show destinations

To show the unicast destinations that are forwarded through each specified egress port, use the **show destinations** command.

```
show destinations [switches | ports] [guids]
```

| Syntax Description |                 |                                                                |
|--------------------|-----------------|----------------------------------------------------------------|
|                    | <i>switches</i> | (Optional) Shows destinations from the specified switch ports. |
|                    | <i>ports</i>    | (Optional) Shows destinations from the specified ports.        |
|                    | <i>guids</i>    | (Optional) Shows GUIDs for each destination.                   |

**Defaults** If no port is specified, then the default is to show all ports.

**Examples** The following is sample output from the **show destinations** command, and it shows all unicast destinations that egress chip switchA/1/3 through port 1:

```
fact> show destinations switchA/1:3/1
```

```
Egress port switchA/1:3/1:
destination lid

switchA/1:2/0 16
host01/1/1 792
```

**Related Commands**

- [show egress ports](#)
- [show multicast destinations](#)
- [show multicast egress ports](#)
- [show multicast routes](#)
- [show routes](#)

# show egress ports

To show the egress ports that forward unicast traffic to each specified destination port, use the **show egress ports** command.

```
show egress ports [chassis | chip] [{to [hosts | switches | ports]} [guids]]
```

## Syntax Description

|                 |                                                                   |
|-----------------|-------------------------------------------------------------------|
| <i>chassis</i>  | (Optional) Shows only switch chassis egress ports.                |
| <i>chip</i>     | (Optional) Shows the egress port of each InfiniBand switch chip.  |
| <i>to</i>       | (Optional) Identifies the destinations set.                       |
| <i>hosts</i>    | (Optional) Shows only the egress ports to the specified hosts.    |
| <i>switches</i> | (Optional) Shows only the egress ports to the specified switches. |
| <i>ports</i>    | (Optional) Shows only the egress ports to the specified ports.    |
| <i>guids</i>    | (Optional) Shows port GUIDs.                                      |

## Defaults

The default view is the chassis view.

## Examples

The following is sample output from the **show egress ports** command, and it shows all unicast egress ports that route to host01:

```
fact> show egress ports to host01

Destination port host01/1/1, lid 792:
egress-port

switchA/1/3
switchB/1/7
```

## Related Commands

- [show destinations](#)
- [show multicast destinations](#)
- [show multicast egress ports](#)
- [show multicast routes](#)
- [show routes](#)

# show history

To show all scans in the repository, use the **show history** command.

**show history**

**Syntax Description** This command has no arguments or keywords.

**Command Default** No default behavior or values.

**Examples** The following is sample output from the **show history** command, and it shows the type, date, and master Subnet Manager of all scans in the repository. Scan 5 is the current scan:

```
fact> show history
```

| index | current | type         | date                    | subnet-manager |
|-------|---------|--------------|-------------------------|----------------|
| 1     |         | fabric       | 2008-04-17 10:44:02 PDT | host01         |
| 2     |         | tech-support | 2008-04-17 10:44:47 PDT | host01         |
| 3     |         | fabric       | 2008-04-17 10:49:01 PDT | host01         |
| 4     |         | fabric       | 2008-04-17 10:49:08 PDT | host01         |
| 5     | Y       | tech-support | 2008-04-17 10:49:39 PDT | host01         |

**Related Commands** [select](#)

[delete scans](#)

# show isolated managed-nodes

To find switches and hosts that are not under the control of a Subnet Manager, use the **show isolated managed-nodes** command.

## **show isolated managed-nodes**

**Syntax Description** This command has no arguments or keywords.

**Defaults** No default behavior or values.

**Usage Guidelines** A managed node will show up in the output of the **show isolated managed-nodes** command if any of the IB chips within the managed node are isolated from the IB fabric. For example, if a host managed node contains two HCAs, and one of those HCAs is isolated from the IB fabric, then that host will be included in the **show isolated managed-nodes** command output.

**Examples** The following is sample output from the **show isolated managed-nodes** command, and it shows that host02 is isolated from the Subnet Manager:

```
fact> show isolated managed-nodes

managed-node-name

host02
```

**Related Commands** [show chassis](#)  
[show subnet-managers](#)

# show multicast destinations

To show the multicast destinations that are forwarded through each specified egress port, use the **show multicast destinations** command.

**show multicast destinations** [*switches* | *ports*] [*to multicast-groups*]

| Syntax Description      |            |                                                                  |
|-------------------------|------------|------------------------------------------------------------------|
| <i>switches</i>         | (Optional) | Shows only the destinations from the specified switch ports.     |
| <i>ports</i>            | (Optional) | Shows only the destinations from the specified ports.            |
| <i>to</i>               | (Optional) | Identifies the destination multicast group set.                  |
| <i>multicast-groups</i> | (Optional) | Shows only the destinations from the specified multicast groups. |

**Defaults** If no port is specified, the default view is the port view. If *multicast-groups* is not specified, all multicast destinations forwarded through each specified egress port are shown.

**Examples** The following is sample output from the **show multicast destinations** command, and it shows all multicast destinations that egress chip switchB/1:1 through port 7:

```
fact> show multicast destinations switchB/1:1/7

Egress port switchB/1/7:
mgid mlid

ff:12:40:1b:ff:ff:00:00:00:00:00:00:00:ff:ff:ff:ff 49152
ff:12:40:1b:ff:ff:00:00:00:00:00:00:00:00:00:01 49153
```

**Related Commands**

- [show egress ports](#)
- [show multicast destinations](#)
- [show multicast egress ports](#)
- [show multicast routes](#)
- [show routes](#)

# show multicast egress ports

To show the egress ports that forward multicast traffic to each specified destination port, use the **show multicast egress ports** command.

**show multicast egress ports** [chassis | chip] [to *multicast-groups* ] [*guids*]

## Syntax Description

|                         |                                                                           |
|-------------------------|---------------------------------------------------------------------------|
| chassis                 | (Optional) Shows only switch chassis egress ports.                        |
| chip                    | (Optional) Shows only the egress ports for each InfiniBand switch chip.   |
| to                      | (Optional) Identifies the destination set.                                |
| <i>multicast-groups</i> | (Optional) Shows only the egress ports to the specified multicast groups. |
| <i>guids</i>            | (Optional) Shows device GUIDs.                                            |

## Defaults

The default is chassis view. If *multicast-groups* is not specified, egress ports for all multicast destinations are shown.

## Examples

The following is sample output from the **show multicast egress ports** command, and it shows the egress ports used for each multicast group:

```
fact> show multicast egress ports

MGID ff:12:40:1b:ff:ff:00:00:00:00:00:00:00:00:01, lid 49153:
egress-port

switchB/1/7

MGID ff:12:40:1b:ff:ff:00:00:00:00:00:00:ff:ff:ff:ff, lid 49152:
egress-port

switchB/1/7

MGID ff:18:a0:1b:00:00:00:00:00:00:05:ad:00:02:80:da, lid 49154:
none found
```

## Related Commands

[show destinations](#)  
[show egress ports](#)  
[show multicast destinations](#)  
[show multicast routes](#)  
[show routes](#)



# show multicast routes

To show multicast routes from each port in the source set to each port in the destination set, use the **show multicast routes** command.

```
show multicast routes [chassis | chip] [{from [src-hosts | src-switches | src-ports | LIDs]}] [{to
dst-hosts | dst-switches | dst-ports}] [multicast-groups] [guids]
```

| Syntax Description |                                                                             |
|--------------------|-----------------------------------------------------------------------------|
| chassis            | (Optional) Shows a chassis-level view of each route.                        |
| chip               | (Optional) Shows a chip-level view of each route.                           |
| from               | (Optional) Shows only the routes from the specified sources.                |
| src-hosts          | (Optional) Shows only the routes from the specified source host ports.      |
| src-switches       | (Optional) Shows only the routes from the specified source switch ports.    |
| src-ports          | (Optional) Shows only the routes from the specified source ports.           |
| to                 | (Optional) Shows only the routes to the specified destinations.             |
| dst-hosts          | (Optional) Shows only the routes to the specified destination host ports.   |
| dst-switches       | (Optional) Shows only the routes to the specified destination switch ports. |
| dst-ports          | (Optional) Shows only the routes to the specified destination ports.        |
| multicast-groups   | (Optional) Shows only the routes for the specified multicast groups.        |
| guids              | (Optional) Shows the GUIDs for each hop on the route.                       |

**Defaults** The default view is the chassis view.

**Examples** The following is sample output from the **show multicast routes** command, and it shows the multicast routes from host01 to host02. It shows two multicast groups routed through different ports between switchA and switchB:

```
fact> show multicast routes from host01 to host02

Route host01/1/1 to host02/1/1,
MGID ff:18:a0:1b:00:00:00:00:00:00:05:ad:00:02:80:da, MLID 49152:
hop chassis ingress-port egress-port

1 switchA 1/1 1/16
2 switchB 1/22 1/1

Route host01/1/1 to host02/1/1,
MGID ff:18:a0:1b:00:00:00:00:00:00:05:ad:00:02:80:db, MLID 49153:
hop chassis ingress-port egress-port

1 switchA 1/1 1/14
2 switchB 1/24 1/1
```

**Related Commands**

- [show destinations](#)
- [show egress ports](#)
- [show multicast destinations](#)

**show multicast egress ports**

**show routes**

# show ports

To show the state of the ports in a given switch or an entire network, use the **show ports** command. (See [Table 5-1 on page 5-2.](#))

```
show ports {chassis | chips | ports} [internal | external] [disabled]
 {state [down | initialize | armed | active]} [summary | guides | detail | full]
```

## Syntax Description

|            |                                                                             |
|------------|-----------------------------------------------------------------------------|
| chassis    | Shows all ports in one or more chassis.                                     |
| chips      | Shows all ports on one or more chips that are specified.                    |
| ports      | Shows one port or multiple ports that you specify.                          |
| internal   | (Optional) Shows the internal ports only.                                   |
| external   | (Optional) Shows the external ports only.                                   |
| disabled   | (Optional) Shows only the ports that are disabled.                          |
| state      | Initializes the port state.                                                 |
| down       | ((Optional) Shows which ports are in the down state.                        |
| initialize | (Optional) Shows which ports are in the initialized state.                  |
| armed      | (Optional) Shows which ports are in the armed state.                        |
| active     | (Optional) Shows which ports are in the active state.                       |
| summary    | (Optional) Shows one line of information for each port.                     |
| guids      | (Optional) Shows one line of information for the GUID and LID of each port. |
| detail     | (Optional) Shows the most important attributes for each port.               |
| full       | (Optional) Shows multiple attributes about each port.                       |

## Command Default

The default view is the summary view, which shows one line per port. If there is no current scan or the current scan is incomplete, using the **show chassis** command automatically performs a fabric scan.

## Usage Guidelines

Shows the port and the neighbor of each port. When you use the GUID view, FACT shows the port GUID and the neighbor GUID. Host ports have GUIDs, so the GUIDS will show, but switch ports do not have GUIDS, so they will not show. If the port is not connected, FACT looks through previous scans for the last-known neighbor and prints the port name and GUID of the neighbor (if it has a GUID), both in square brackets, and it also shows the scan number.

If the port was connected yet is no longer connected, the port will have a last-known neighbor. The last-known neighbor appears in square brackets, followed by the scan number in which it was last seen.

## Examples

The following is sample output from the **show ports** command, and it shows all active external ports in the subnet:

```
fact> show ports external state active
```

| port        | state  | capacity | neighbor    |
|-------------|--------|----------|-------------|
| host01/1/1  | active | 4x-ddr   | switchB/1/7 |
| switchA/1/3 | active | 4x-sdr   | switchB/1/3 |
| switchB/1/3 | active | 4x-sdr   | switchA/1/3 |
| switchB/1/7 | active | 4x-ddr   | host01/1/1  |

---

**Related Commands**

[show changes scan](#)

[show chassis](#)

[show chips](#)

[show multicast groups](#)

# show routes

To show the unicast routes from each port in the source set to each port in the destination set, use the **show routes** command.

```
show routes [chassis | chip] {from [src-hosts | src-switches | src-ports]} {to [dst-hosts | dst-switches | dst-ports]} [guids]
```

## Syntax Description

|                     |                                                            |
|---------------------|------------------------------------------------------------|
| chassis             | (Optional) Shows a chassis-level view of each route.       |
| chip                | (Optional) Shows a chip-level view of each route           |
| from                | (Optional) Indicates from which port the route originates. |
| <i>src-hosts</i>    | (Optional) Identifies the hosts from the source set.       |
| <i>src-switches</i> | (Optional) Identifies the switches from the source set.    |
| <i>src-ports</i>    | (Optional) Identifies the ports from the source set.       |
| to                  | (Optional) Identifies the destination set.                 |
| <i>dst-hosts</i>    | (Optional) Identifies the hosts in the destination set.    |
| <i>dst-switches</i> | (Optional) Identifies the switches in the destination set. |
| <i>dst-ports</i>    | (Optional) Identifies the ports in the destination set.    |
| guids               | (Optional) Shows device GUIDs.                             |

## Command Default

If the chassis or chip are not specified, the default view is the chassis-level view. If the source or the destination are not specified, then the default is to view the routes from all sources to all destinations.

## Examples

The following is sample output from the **show routes** command, and it shows the chassis view of all routes from switchA to host01. Each route has two hops. SwitchA has three switch chips, and port 0 on each switch chip is a source.

```
fact> show routes from switchA to host01

Route switchA/1:1/0 to host01/1/1, LID 792:
hop chassis ingress-port egress-port

1 switchA 1:1/0 1/3
2 switchB 1/3 1/7

Route switchA/1:2/0 to host01/1/1, LID 792:
hop chassis ingress-port egress-port

1 switchA 1:2/0 1/3
2 switchB 1/3 1/7

Route switchA/1:3/0 to host01/1/1, LID 792:
hop chassis ingress-port egress-port

1 switchA 1:3/0 1/3
2 switchB 1/3 1/7
```

The following example shows the chip view of all routes from switchA to host01. SwitchA has three switch chips, and port 0 on each switch chip is a source. Packets from switchA/1:1/0 and switchA/1:3/0 route through chip switchA/1:2.

```
fact> show routes chip from switchA to host01
```

```
Route switchA/1:1/0 to host01/1/1, LID 792:
hop chip ingress-port egress-port

1 switchA/1:1 0 4
2 switchA/1:2 1 8
3 switchB/1:1 3 7

Route switchA/1:2/0 to host01/1/1, LID 792:
hop chip ingress-port egress-port

1 switchA/1:2 0 8
2 switchB/1:1 3 7

Route switchA/1:3/0 to host01/1/1, LID 792:
hop chip ingress-port egress-port

1 switchA/1:3 0 1
2 switchA/1:2 3 8
3 switchB/1:1 3 7
```

#### Related Commands

[show destinations](#)

[show egress ports](#)

[show multicast destinations](#)

[show multicast egress ports](#)

[show multicast routes](#)

# show subnet-managers

To all the subnet managers in the current scan, use the **show subnet-managers** command.

**show subnet-managers**

**show sm**

## Syntax Description

This command has no arguments or keywords.

## Usage Guidelines

Unlike most commands, the **show subnet-managers** command works on the current scan if the scan is not complete.

## Examples

The following is sample output from the **show subnet-managers** command, and it shows the subnet manager in the current scan. Host01 is the master Subnet Manager, and switchA and switchB are standby Subnet Managers:

```
fact> show subnet-managers
```

```
subnet-prefix master standby (* for non-active)

fe:80:00:00:00:00:00 host01 switchA
 switchB
 switchC *
```

## Related Commands

[scan fabric](#)

[show isolated managed-nodes](#)

# show versions

To show the version of the firmware on each managed node, use the **show versions** command.

**show versions** [*managed-nodes*]

## Syntax Description

*managed-nodes* (Optional) Displays only the versions of the specified managed nodes.

## Command Default

No default behavior or values.

## Usage Guidelines

Unlike most commands, the **show versions** command works on the current scan, even if the scan is not complete.

The **show versions** command displays different version information depending upon your system hardware and software:

- For hosts running the Cisco IB stack, the version displayed is the version of the installed ib-mod RPM.
- For hosts running the OFED stack, the version displayed is the version reported by ofed\_info, which should be the OFED driver release.
- For SFS OS switches, the version displayed is the system version reported by the **show versions** command.
- For OEM switches, the version displayed is the version reported by the **fwVersion** command.

## Examples

The following is sample output from the **show versions** command, and it shows the versions of all managed nodes in the subnet:

```
fact> show versions

name version

host01 Cisco rhel4-2.6.9-42.ELsmp-3.2.0-148
switchA Topspin-90 SFS OS 2.10.0-ALPHA releng #613 11/05/2007 21:36:18
switchB SFS-7000D SFS OS 2.11.0-ALPHA releng #91 03/02/2008 00:07:09
```

## Related Commands

[install](#)

# Command Groups

This section lists the FACT commands, grouped by function for those who prefer a structured organization. For a complete description of all commands in an alphabetical listing, see the “[Commands](#)” section on page 5-10.

Table 5-5 lists the commands that scan the managed-nodes of the network.

**Table 5-5 Scanning Commands**

| Command Name             | Function                                                                                                                                                                                                                 |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>scan fabric</b>       | Scans the static connectivity of the subnet and collects information from and about the Subnet Managers.                                                                                                                 |
| <b>scan fabric all</b>   | Scans the static connectivity of the subnet and collects routing information, raw port counter information, monitored port counter information, and threshold-exceeded port counter information from the Subnet Manager. |
| <b>scan tech-support</b> | Collects diagnostic information about the network.                                                                                                                                                                       |
| <b>select</b>            | Selects the current scan.                                                                                                                                                                                                |

Table 5-6 lists the commands that manage the repository.

**Table 5-6 Repository Management Commands**

| Command Name             | Function                                                                                    |
|--------------------------|---------------------------------------------------------------------------------------------|
| <b>delete scans</b>      | Deletes all specified scans.                                                                |
| <b>export repository</b> | Makes a copy of the specified repository and saves a copy of the FACT repository in a file. |
| <b>export scan</b>       | Exports the current scan to a file.                                                         |
| <b>import repository</b> | Replaces the existing repository with the one that is currently in the file.                |
| <b>import scan</b>       | Imports a scan from a file to the repository.                                               |
| <b>select</b>            | Selects the current scan.                                                                   |
| <b>show history</b>      | Shows all scans in the repository.                                                          |

Table 5-7 lists the commands that work with any scan, even an incomplete scan.

**Table 5-7 Low-Level Display Commands**

| Command Name                | Function                                                |
|-----------------------------|---------------------------------------------------------|
| <b>show subnet-managers</b> | Shows the Subnet Managers in the current scan.          |
| <b>show versions</b>        | Shows the scan versions of the specified managed-nodes. |

Table 5-8 lists the commands that show information about any scan.

**Table 5-8 Subnet-Level Display Commands**

| Command Name                       | Function                                                             |
|------------------------------------|----------------------------------------------------------------------|
| <b>show chassis</b>                | Shows information about the chassis in the current scan.             |
| <b>show chips</b>                  | Shows information about switch chips in the current scan.            |
| <b>show isolated managed-nodes</b> | Shows the switches and hosts that are not part of the network.       |
| <b>show multicast groups</b>       | Shows the multicast groups in the current scan.                      |
| <b>show ports</b>                  | Shows the state of the ports in a given switch or an entire network. |

Table 5-9 lists the commands that show information about a route scan.

**Table 5-9 Route-Level Display Commands**

| Command Name                       | Function                                                                                       |
|------------------------------------|------------------------------------------------------------------------------------------------|
| <b>show destinations</b>           | Shows the unicast destinations that are forwarded through each specified egress port.          |
| <b>show egress ports</b>           | Shows the egress ports that forward unicast traffic to each specified destination port.        |
| <b>show multicast destinations</b> | Shows the multicast destinations that are forwarded through each specified egress port.        |
| <b>show multicast egress ports</b> | Shows the egress ports that forward multicast traffic to each specified destination port.      |
| <b>show routes</b>                 | Shows the unicast routes from each port in the source set to each port in the destination set. |

Table 5-10 lists the commands that show port counters.

**Table 5-10 Port Counters Display Commands**

| Command Name                   | Function                                                           |
|--------------------------------|--------------------------------------------------------------------|
| <b>show counters monitored</b> | Shows monitored port counters.                                     |
| <b>show counters raw</b>       | Shows raw port counters.                                           |
| <b>show counters threshold</b> | Shows the port counters that have exceeded their error thresholds. |

Table 5-11 lists the command that compares two scans.

**Table 5-11 History Display Commands**

| Command Name             | Function                                                             |
|--------------------------|----------------------------------------------------------------------|
| <b>show changes scan</b> | Compares the current scan with any specified scan in the repository. |

Table 5-12 lists the commands that are used to administer or manage the network.

**Table 5-12 Fabric Maintenance Commands**

| Command Name               | Function                                                                  |
|----------------------------|---------------------------------------------------------------------------|
| <b>disable ports</b>       | Disables one or more ports on a switch.                                   |
| <b>enable ports</b>        | Enables one or more ports on a switch.                                    |
| <b>install</b>             | Installs new firmware on a switch.                                        |
| <b>ports speed</b>         | Sets the width and speed of a single port or multiple ports in a network. |
| <b>reset port-counters</b> | Resets all port counters to zero.                                         |

Table 5-13 lists the commands used to analyze, display, annotate information in log files.

**Table 5-13 Log Analysis Commands**

| Command Name    | Function                                                                                    |
|-----------------|---------------------------------------------------------------------------------------------|
| <b>annotate</b> | Annotates one or more syslog files and prints them to the screen with annotations attached. |

Table 5-14 lists the miscellaneous commands.

**Table 5-14 Miscellaneous Commands**

| Command Name | Function                                              |
|--------------|-------------------------------------------------------|
| <b>exit</b>  | Discontinues the current session with FACT.           |
| <b>help</b>  | Lists a summary of the complete FACT command grammar. |





## Acronyms and Abbreviations

---

[Table A-1](#) includes the expanded acronyms and abbreviations used throughout this *Cisco Fabric Analysis and Correlation Toolkit User Guide*.

**Table A-1**      *Expanded Acronyms and Abbreviations*

| Acronym | Expansion                                      |
|---------|------------------------------------------------|
| CLI     | command-line interface                         |
| DNS     | Domain Name System                             |
| FACT    | Fabric Analysis and Correlation Toolkit        |
| GID     | global identifier                              |
| GUID    | global unique identifier                       |
| HCA     | host channel adapter                           |
| HSM     | High-Performance Subnet Manager                |
| IB      | InfiniBand                                     |
| LID     | local identifier                               |
| MLID    | multicast locally unique identifier            |
| OEM     | Original Equipment Manufacturer                |
| OFED    | Open Fabrics Enterprise Distribution           |
| P_Keys  | partition keys                                 |
| PM      | Performance Manager                            |
| SFS     | Server Fabric Switch                           |
| SFS OS  | Server Fabric Switch operating system software |
| SM      | Subnet Manager                                 |
| SMA     | Subnet Management Agent                        |
| SSH     | secure shell                                   |
| TAC     | Technical Assistance Center                    |





## INDEX

---

### A

- acronyms and abbreviations list [A-1](#)
- annotation
  - using [4-4](#)

---

### C

- checklist, installation and configuration [3-1](#)

#### CLI

- editing [5-8](#)
- environment variables [5-7](#)
- options [5-5](#)

#### command modes

- about [5-4](#)
- interactive mode [5-4](#)
- single-command mode [5-5](#)

#### commands

- correcting [5-8](#)
- grouped by function [5-10](#)
- list of [5-10](#)

#### credentials files

- about [3-5](#)
- creating [3-12](#)

---

### D

#### documentation

- audience [i-v](#)
- conventions [i-vi](#)
- related [i-vi](#)

---

### F

#### FACT

- about [1-1](#)
- configuring [3-10](#)
- connections [1-2](#)
- how it works [1-1](#)
- quick configuration [2-1](#)

#### firmware

- installing [4-6](#)
  - in managed switches [4-6](#)
  - in unmanaged switches [4-7](#)

---

### G

#### GUID name definitions

- about [3-9](#)
- creating [3-11](#)

---

### M

#### managed node definitions files

- about [3-4](#)
- creating [3-11](#)

#### master configuration files

- about [3-3](#)
- configuring [3-10](#)

---

### P

#### port counters

- commands [4-6](#)
- using [4-5](#)

---

## R

### repository

- about [1-2](#)
- maintaining [4-2](#)
- management commands [4-3](#)

---

## S

### scanning

- about [1-2](#)
- command list [4-2](#)
- types of scans [4-1](#)
  - fabric scan [4-1](#)
  - incomplete scan [4-1](#)
  - port-counters scan [4-1](#)
  - tech-support scan [4-1](#)
- using [4-1](#)

### Secure Shell

- understanding [1-4](#)

### Subnet Manager

- understanding [1-4](#)

---

## T

- troubleshooting [2-4](#)

---

## W

- wildcard values [3-7](#)