

Contents

- 1 Using FACT Command Arguments
 - ◆ 1.1 Using Singular Arguments
 - ◆ 1.2 Using Plural Arguments
- 2 Using Command Modes
 - ◆ 2.1 Using Interactive Mode
 - ◆ 2.2 Using Single-Command Mode
- 3 Command-line Interface
 - ◆ 3.1 Command-line Options
 - ◆ 3.2 Environment Variables
 - ◆ 3.3 Correcting Commands
 - ◆ 3.4 Editing the CLI

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.

Using Singular Arguments

This section describes the syntax for each type of argument. [Table 1](#) lists the argument syntax and the corresponding descriptions.

Table 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 show history 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"> • hostname or IP address of its management port, as specified in the managed-node definition. See About Managed Node Definitions Files. • chassis GUID, using the notation 00:11:22:33:44:55:66:77. • given name of its chassis GUID (from guid name definition).

Using_the_CLI

	<ul style="list-style-type: none"> • description of one of its chips, prepended by ?chassis,? as shown in the following example: chassis:00:11:22:33:44:55:66:77 (a node GUID)
<i>host</i>	<p>A host can be specified in one of the following ways:</p> <ul style="list-style-type: none"> • hostname or IP address, as specified in the managed-node definition. • given name of its HCA node GUID (from guid name definitions). • node GUID of its HCA, prepended by ?chassis.?
<i>chassis</i>	<p>A chassis is either a host or a switch. Any of the previously listed notations for host or switch may be used.</p>
<i>managed-node</i>	<p>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 About Managed Node Definitions Files.</p>
<i>chip</i>	<p>A chip is an InfiniBand switch chip or InfiniBand channel adapter chip. It may be specified in the following ways:</p> <ul style="list-style-type: none"> • node GUID. • given name from the GUID name definitions. • 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.
<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> <p style="text-align: center;">00:02:c9:02:00:22:2a:c8</p> <p>Note: Most switch chips do not have port GUIDs.</p> <p>Chassis-relative form has the <i>'syntax 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> <p style="text-align: center;">switch01/2/3 (port 3 on slot 2 of chassis switch01)</p> <p>Note: Internal chips can be specified in chip-relative syntax only.</p> <p>Chip-relative form has the <i>syntax 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> <p style="text-align: center;">switch01/2:3/4</p>

Using_the_CLI

	(chip port 4 on chip 3 of the card in slot 2 of chassis switch01) 00:02:c9:02:00:22:2a:c4/4 (chip port 4 on the chip with the given GUID)
<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

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
-----
switch01                          TopspinOS 2.9.0 releng #147 10/25/...
switch02                          TopspinOS 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
name                               version
-----
switch01                          TopspinOS 2.9.0 releng #147 10/25/...
switch02                          TopspinOS 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 shows how to use it.

Command-line Options

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

Table 2: Command-line Options

Option	Description
-c <i>file</i> --config-file= <i>file</i>	Reads primary configuration from the specified file.
--color=always	If tracing, specifies that the transcript is colorized, showing FACT requests in one color and the switch responses in another.
--color=auto	If tracing, specifies that the transcript is colorized only if the FACT standard output is a terminal. This behavior is the default.
--color=never	If tracing, specifies that the transcript is not colorized.
--config-option= <i>option</i>	Adds or overrides a configuration option.
--credentials-from= <i>file</i>	Adds credential information from the file. See About Credentials Files .
--guid-name= <i>guid=name</i>	Adds a GUID name definition. See About GUID Name Definitions section.
--guid-names- from= <i>file</i>	Adds GUID name definitions from the specified file. See About GUID Name Definitions .
-h --help	Prints a message describing messages.
--help-commands	Lists FACT commands.

--managed-nodes=<i>type: pattern</i>	Adds a set of managed-nodes to those about which FACT is aware. See About Managed Node Definitions Files .
--managed-nodes-from=<i>file</i>	Adds managed-nodes from the file. See About Managed Node Definitions Files .
-s --sequential	Initiates sequential scan sessions to one managed node at a time. This option is helpful with the --trace command because output displays slower and more predictably. By default, FACT connects up to 50 managed nodes at a time during scans and installations. If sequential is specified, then FACT connects to one managed node at a time.
--subnet-manager=<i>domain name or IP address</i>	Specifies the InfiniBand Subnet Manager host that FACT should use.
--trace and --sequential	Runs sequential sessions to managed nodes and prints the session transcript.
-t --trace	By default, if --trace is not specified, then FACT will print the transcript of each session when that session is closed. If --trace is specified, FACT will print the transcript of each session as it occurs.
-v --version	Prints FACT version information and exits.

Environment Variables

FACT uses environment variables to control the FACT configuration. [Table 3](#) lists and describes the FACT environment variables.

Table 3: Environment Variables

Variable	Description
FACT_CONF	Name of master configuration file. See About Master Configuration Files .
FACT_CREDENTIAL_FILES	List of files that contain credentials, separated by colons, as shown in the following: FACT_CREDENTIAL_FILES=/etc/fact/cred1:/etc/fact/cred2 states that the files /etc/fact/cred1 and /etc/fact/cred2 contain GUID names. See About Credentials Files .
FACT_GUID_NAMES	

Using the CLI

	<p>List of GUID name definitions, separated by semicolons. Each definition has the form <code>?GUID=name,?</code> as shown in the following example:</p> <pre>FACT_GUID_NAMES="00:1b:0d:0b:00:00:ca:a6=switchA;00:1b:0d:0b:00:00:ca:a6=switchB;00:1b:0d:0b:00:00:ca:aa"</pre> <p>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</p> <p>See About GUID Name Definitions.</p>
FACT_GUID_NAME_FILES	<p>List of files containing GUID name definitions, separated by colons, as shown in the following example:</p> <pre>FACT_GUID_NAME_FILES=/tmp/guidnames1:/tmp/guidnames2</pre> <p>states that the files /tmp/guidnames1 and /tmp/guidnames2 contain GUID name definitions</p> <p>See About GUID Name Definitions.</p>
FACT_MANAGED_NODES	<p>In-line list of managed node definitions, separated by colons, as shown in the following example:</p> <pre>FACT_MANAGED_NODES="host=ibmg-r1-evil:SFSOS switch=209.16.209.16"</pre> <p>defines a host named ibmg-r1-evil and an SFS OS switch named 209.16.209.16</p> <p>See About Managed Node Definitions Files.</p>
FACT_MANAGED_NODE_FILES	<p>List of files containing managed node definitions, separated by colons, as shown in the following example:</p> <pre>FACT_MANAGED_NODE_FILES=/tmp/nodefile1:/tmp/nodefile2</pre> <p>states that the files /tmp/nodefile1 and /tmp/nodefile2 contain managed node definitions</p> <p>See About Managed Node Definitions Files.</p>

Correcting Commands

Note: This feature will be functional at FCS.

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

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 4](#) lists and describes these options.

Table 4: Key Stroke Shortcuts

Key Strokes	Description
Ctrl-A	Moves the cursor to the beginning of the line.
Ctrl-B	Moves the cursor to the left (backward) one character.
Ctrl-D	Deletes the current character. 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 the Backspace key.
Ctrl-I	Completes the current word, analogous to the Tab key.
Ctrl-J	Accepts the current line.
Ctrl-K	Deletes text from the cursor to the end of the line.
Ctrl-L	Refreshes the input line.
Ctrl-M	Accepts the current line, analogous to the 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 the FACT.
Ctrl-Del	Deletes the character to the left of the cursor, analogous to the Delete key.
?	Completes the current word.
Esc-[A	Moves to the previous line in the command history.

Using_the_CLI

Esc-[B	Proceeds to the next line in the command history.
Esc-[C	Moves the cursor to the right (forward) one character.
Esc-[D	Moves the cursor to the left (backward) on character.
Esc-B	Moves the cursor to the left (backward) one word.
Esc-C	Capitalizes the word to the right of the cursor (forward).
Esc-D	Deletes characters from the cursor through the 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 word 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.