

FACT uses four types of configuration files that you must create. This section describes the files.

Contents

- [1 About Master Configuration Files](#)
- [2 About Managed Node Definitions Files](#)
- [3 About Credentials Files](#)
 - ◆ [3.1 Device Types](#)
 - ◇ [3.1.1 Table 1: Device Types](#)
 - ◆ [3.2 Wildcard Values](#)
 - ◇ [3.2.1 Table 2: Wildcard Values](#)
 - ◆ [3.3 Name-Value Pairs](#)
 - ◇ [3.3.1 Table 3: Parameters for Name-Value Pairs](#)
 - ◆ [3.4 About GUID Name Definitions](#)

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 [About Credentials Files](#).)

- `managed-nodes: type pattern`

A managed-node definition. (See [About Managed Node Definitions Files](#).)

- `managed-node-file: file`

A file containing managed node definitions. (See [About Managed Node Definitions Files](#).)

About_FACT_Configuration_Files

- `guid-name`: *guid name*

A GUID name definition. (See [About GUID Name Definitions](#).)

- `guid-names-file`: *file*

A file containing GUID name definitions. (See [About GUID Name Definitions](#).)

- `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 [Creating a Master Configuration File](#).

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.

About_FACT_Configuration_Files

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 [Creating a Separate Managed Node Definitions File](#).

About Credentials Files

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 [Creating a Credentials File](#).

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 1](#) lists the device types that may appear in the device type line, and [Table 2](#) lists the wildcard values.

Table 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 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 wildcards can be either letters or numbers.

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

Table 3: Parameters for Name-Value Pairs

Name-Value Pair	Description
user	Login username. The default is "super" on SFS OS switches and "admin" on OEM switches. No default exists on the host.

About_FACT_Configuration_Files

	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" to "direct." The default is "direct for the localhost and "SSH" for others. Most users do not use this parameter.
port	TCP port number to user 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 the default identity file, yet, SSH defaults to ~/.ssh/id_rsa or id_dsa. Most Users do not use this parameter.
HSM-command	The executable program to invoke the HSM CLI. The default is /usr/local/topspin/sbin/ib_sm_cli or /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 vstate 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 SSH to connect to the managed node. The direct method is only used to scan the host upon which FACT is running. (See [Understanding Secure Shell](#) for more information about SSH.)

The following example shows a possible credentials file:

```
# This is a comment.
SFSOS switch switch4*
  user: jsmith
  password: t0psekr1t
host hsm-[1-3]
  method: ssh
  user: fact
  ssh-identity: ~/.ssh/id_rsa
```

Table 3: Parameters for Name-Value Pairs

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 host names 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 [Creating a GUID Name Definition File](#).