



*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*



## Cisco Fabric Manager Web Services Programming Guide

Cisco MDS NX-OS Release 4.2(0)  
Cisco MDS 9000 FabricWare Release 4.x  
August 2009

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

Text Part Number: OL-19780-01

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.

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, CCSI, Cisco Eos, Cisco HealthPresence, Cisco IronPort, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco Nurse Connect, Cisco StackPower, Cisco StadiumVision, Cisco TelePresence, Cisco Unified Computing System, Cisco WebEx, DCE, Flip Channels, Flip for Good, Flip Mino, Flip Video, Flip Video (Design), Flipshare (Design), Flip Ultra, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn, Cisco Store, and Flip Gift Card are service marks; 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, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, 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. (0907R)

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 Manager Web Services Programming Guide*

© 2009 Cisco Systems, Inc. All rights reserved.



## CONTENTS

### **New and Changed Information** v

#### **Preface** vii

Audience	vii
Organization	vii
Document Conventions	vii
Related Documentation	viii
Release Notes	viii
Regulatory Compliance and Safety Information	viii
Compatibility Information	viii
Hardware Installation	ix
Software Installation and Upgrade	ix
Cisco Fabric Manager	ix
Command-Line Interface	ix
Intelligent Storage Networking Services Configuration Guides	x
Troubleshooting and Reference	x
Obtaining Documentation and Submitting a Service Request	x

---

#### CHAPTER 1

### **Web Services Overview** 1-1

---

#### CHAPTER 2

### **Fabric Manager Web Services** 2-1

About Fabric Manager Web Services	2-1
Web Services Specifications	2-1
XML	2-2
SOAP	2-2
HTTP/HTTPS	2-2
WDSL	2-2
Logon Service	2-2
<b>requestToken</b>	2-2
<b>validateToken</b>	2-3
Authentication or Token	2-3
IdentityManager	2-3
San Service	2-4
Service Endpoint Interface (SEI)	2-4

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

Methods	2-4
<a href="#">getFabrics</a>	2-4
<a href="#">getFabricByIP</a>	2-4
<a href="#">getFabricByKey</a>	2-4
<a href="#">getFabricBySwitchKey</a>	2-5
<a href="#">getSwitchesByFabric</a>	2-5
<a href="#">getSwitch</a>	2-5
<a href="#">getSwitchByKey</a>	2-6
<a href="#">getSwitchIPByName</a>	2-6
<a href="#">getSwitchIPByKey</a>	2-6
<a href="#">getNeighborSwitches</a>	2-7
<a href="#">getVsans</a>	2-7
<a href="#">getVsan</a>	2-7
<a href="#">getIsIs</a>	2-8
<a href="#">discoverFabric</a>	2-8
<a href="#">manageFabric</a>	2-8
<a href="#">unManageFabric</a>	2-9
<a href="#">closeFabric</a>	2-9
<a href="#">purgeFabric</a>	2-9
<a href="#">getEndpoints</a>	2-10
<a href="#">getEnclosures</a>	2-10
<a href="#">getEndPointByKey</a>	2-10
<a href="#">getEndPointAttachedToSw</a>	2-10
<a href="#">getEnclosureByName</a>	2-11
<a href="#">getEnclosureByKey</a>	2-11
<a href="#">getEnclosureByPwwn</a>	2-11
<a href="#">updateEnclosure</a>	2-12
<a href="#">updateEndpointEnclosure</a>	2-12
<a href="#">getHosts</a>	2-12
<a href="#">getHost</a>	2-13
<a href="#">getHostByFabric</a>	2-13
<a href="#">getStorages</a>	2-13
<a href="#">getStorageByFabric</a>	2-13
<a href="#">getHostPorts</a>	2-14
<a href="#">getDomainId</a>	2-14
<a href="#">getVsanIp</a>	2-14
<a href="#">getVsanDomains</a>	2-15
Cluster WS - SEI	2-15
Method	2-15
<a href="#">getSwitchesByFabricKey</a>	2-15

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

<a href="#">getServerIpByFabricKey</a>	2-16
<a href="#">getServerIpBySwitchKey</a>	2-16
<a href="#">getFabricsByServerIp</a>	2-16
<a href="#">getAllServers</a>	2-16
<a href="#">getFabricByEnclosureKey</a>	2-17
<a href="#">getServerIpByEnclosureKey</a>	2-17
<a href="#">getServerIpByVsanKey</a>	2-17
Event WS - SEI	2-17
Method	2-18
<a href="#">isCallHomeEnabled</a>	2-18
<a href="#">getCallHomeDestProfile</a>	2-18
<a href="#">getCallHomeSysInfo</a>	2-18
<a href="#">getEmailMaxEntries</a>	2-18
<a href="#">getEmailSetup</a>	2-19
<a href="#">getSyslogServers</a>	2-19
<a href="#">getSyslogMessageControl</a>	2-19
<a href="#">getSyslogLoggingCfg</a>	2-19
Protocol WS - SEI	2-20
Method	2-20
<a href="#">getNtpPeers</a>	2-20
<a href="#">getNtpInfo</a>	2-20
<a href="#">getFspfConfig</a>	2-20
<a href="#">queryInterfaceFspfConfig</a>	2-21
Security WS - SEI	2-21
Method	2-21
<a href="#">getAaaMaxServer</a>	2-21
<a href="#">getAaaMaxAppServer</a>	2-22
<a href="#">isMSCHAPRequired</a>	2-22
<a href="#">getAaaSetup</a>	2-22
<a href="#">getAaaAppServerGroups</a>	2-22
<a href="#">getAaaServerGroups</a>	2-23
<a href="#">getSnmpUsers</a>	2-23
<a href="#">getIPACLProfiles</a>	2-23
<a href="#">getSSHConfig</a>	2-24
<a href="#">getSSHEnabled</a>	2-24
<a href="#">isTelnetEnabled</a>	2-24
<a href="#">getPkiRsaKeys</a>	2-24
<a href="#">getPkiTrustPointNames</a>	2-25
<a href="#">getPkiTrustPointNames</a>	2-25

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

<a href="#">getPkiCert</a>	2-25
<a href="#">getPkiAction</a>	2-25
<a href="#">getPkiTrustPoint</a>	2-26
<a href="#">getFeatureControls</a>	2-26
<a href="#">getIkeFailRecoveryCfg</a>	2-26
<a href="#">getIkeCfgPolicies</a>	2-27
<a href="#">getIkeCfgInitiators</a>	2-27
<a href="#">getIkeTunnels</a>	2-27
<a href="#">getIPsecGlobalCfg</a>	2-27
<a href="#">getIPsecXformSets</a>	2-28
<a href="#">getIPsecCryptoMaps</a>	2-28
<a href="#">getIfsFromCryptoMap</a>	2-28
<a href="#">getIPsecTunnels</a>	2-28
<a href="#">Error Codes</a>	2-29



Cisco Confidential - Draft Review

## New and Changed Information

This document provides release-specific information for each new and changed feature in Cisco MDS Fabric Manager Release 4.x software. The *Cisco MDS 9000 Family Cisco Fabric Manager Web Services Programming Guide* is updated to address each new and changed feature. The latest version of this document is available at the [Cisco MDS 9000 NX-OS Software Configuration Guides](#) website.



### Tip

The configuration guides created for earlier releases are also listed at the aforementioned website. Each guide addresses the features introduced or available in those releases. Select and view the configuration guide pertinent to the software installed in your switch.

To check for additional information about this release, refer to the *Cisco MDS 9000 Family Release Notes* available at the [Cisco MDS 9000 NX-OS Software Release Notes](#) website.

**Table 1** summarizes the new and changed features for the *Cisco MDS 9000 Family Cisco Fabric Manager Web Services Programming Guide*, and tells you where they are documented. The table includes a brief description of each new feature and the release in which the change occurred.

**Table 1** New and Changed Features for Cisco MDS Fabric Manager Release 4.x

Feature	GUI Changes	Description	Changed in Release	Where Documented
Security WS	NA	Security Web Services	4.2(1)	<a href="#">Chapter 2, “Fabric Manager Web Services”</a>
ProtocolWS	NA	Protocol Web Services	4.2(1)	<a href="#">Chapter 2, “Fabric Manager Web Services”</a>
EventWS	NA	Event Web Services	4.2(1)	<a href="#">Chapter 2, “Fabric Manager Web Services”</a>

*Cisco Confidential - Draft Review*





## Preface

---

This preface describes the audience, organization, and conventions of the *Cisco MDS 9000 Family Cisco Fabric Manager Web Services Programming Guide*. It also provides information on how to obtain related documentation.

## Audience

This guide is for experienced network administrators who are responsible for configuring and maintaining the Cisco MDS 9000 Family of multilayer directors and fabric switches.

## Organization

The *Cisco Fabric Manager Web Services Programming Guide* is organized as follows:

Chapter	Title	Description
<a href="#">Chapter 1</a>	<a href="#">Web Services Overview</a>	Provides a brief overview of Fabric Manager Web Services and its applications.
<a href="#">Chapter 2</a>	<a href="#">Fabric Manager Web Services</a>	Describes the Fabric Manager Web Services (FMWS) application program interface (API)

## Document Conventions

Command descriptions use these conventions:

<b>boldface font</b>	Commands and keywords are in boldface.
<i>italic font</i>	Arguments for which you supply values are in italics.
[ ]	Elements in square brackets are optional.
[ x   y   z ]	Optional alternative keywords are grouped in brackets and separated by vertical bars.

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

Screen examples use these conventions:

<code>screen font</code>	Terminal sessions and information the switch displays are in screen font.
<b>boldface screen font</b>	Information you must enter is in boldface screen font.
<i>italic screen font</i>	Arguments for which you supply values are in italic screen font.
< >	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.

This document uses the following conventions:



**Note**

Means reader *take note*. Notes contain helpful suggestions or references to material not covered in the manual.



**Caution**

Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

## Related Documentation

The documentation set for the Cisco MDS 9000 Family includes the following documents. To find a document online, use the Cisco MDS NX-OS Documentation Locator at:

[http://www.cisco.com/en/US/docs/storage/san\\_switches/mds9000/roadmaps/doclocator.htm](http://www.cisco.com/en/US/docs/storage/san_switches/mds9000/roadmaps/doclocator.htm)

## Release Notes

- *Cisco MDS 9000 Family Release Notes for Cisco MDS NX-OS Releases*
- *Cisco MDS 9000 Family Release Notes for MDS SAN-OS Releases*
- *Cisco MDS 9000 Family Release Notes for Storage Services Interface Images*
- *Cisco MDS 9000 Family Release Notes for Cisco MDS 9000 EPLD Images*
- *Release Notes for Cisco MDS 9000 Family Fabric Manager*

## Regulatory Compliance and Safety Information

- *Regulatory Compliance and Safety Information for the Cisco MDS 9000 Family*

## Compatibility Information

- *Cisco Data Center Interoperability Support Matrix*
- *Cisco MDS 9000 NX-OS Hardware and Software Compatibility Information and Feature Lists*

[Send documentation comments to mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)

- *Cisco MDS NX-OS Release Compatibility Matrix for Storage Service Interface Images*
- *Cisco MDS 9000 Family Switch-to-Switch Interoperability Configuration Guide*
- *Cisco MDS NX-OS Release Compatibility Matrix for IBM SAN Volume Controller Software for Cisco MDS 9000*
- *Cisco MDS SAN-OS Release Compatibility Matrix for VERITAS Storage Foundation for Networks Software*

## Hardware Installation

- *Cisco MDS 9500 Series Hardware Installation Guide*
- *Cisco MDS 9200 Series Hardware Installation Guide*
- *Cisco MDS 9100 Series Hardware Installation Guide*
- *Cisco MDS 9124 and Cisco MDS 9134 Multilayer Fabric Switch Quick Start Guide*

## Software Installation and Upgrade

- *Cisco MDS 9000 NX-OS Release 4.1(x) and SAN-OS 3(x) Software Upgrade and Downgrade Guide*
- *Cisco MDS 9000 Family Storage Services Interface Image Install and Upgrade Guide*
- *Cisco MDS 9000 Family Storage Services Module Software Installation and Upgrade Guide*

## Cisco NX-OS

- *Cisco MDS 9000 Family NX-OS Licensing Guide*
- *Cisco MDS 9000 Family NX-OS Fundamentals Configuration Guide*
- *Cisco MDS 9000 Family NX-OS System Management Configuration Guide*
- *Cisco MDS 9000 Family NX-OS Interfaces Configuration Guide*
- *Cisco MDS 9000 Family NX-OS Fabric Configuration Guide*
- *Cisco MDS 9000 Family NX-OS Quality of Service Configuration Guide*
- *Cisco MDS 9000 Family NX-OS Security Configuration Guide*
- *Cisco MDS 9000 Family NX-OS IP Services Configuration Guide*
- *Cisco MDS 9000 Family NX-OS Intelligent Storage Services Configuration Guide*
- *Cisco MDS 9000 Family NX-OS High Availability and Redundancy Configuration Guide*
- *Cisco MDS 9000 Family NX-OS Inter-VSAN Routing Configuration Guide*

## Cisco Fabric Manager

- *Cisco Fabric Manager Fundamentals Configuration Guide*
- *Cisco Fabric Manager System Management Configuration Guide*
- *Cisco Fabric Manager Interfaces Configuration Guide*

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

- *Cisco Fabric Manager Fabric Configuration Guide*
- *Cisco Fabric Manager Quality of Service Configuration Guide*
- *Cisco Fabric Manager Security Configuration Guide*
- *Cisco Fabric Manager IP Services Configuration Guide*
- *Cisco Fabric Manager Intelligent Storage Services Configuration Guide*
- *Cisco Fabric Manager High Availability and Redundancy Configuration Guide*
- *Cisco Fabric Manager Inter-VSAN Routing Configuration Guide*
- *Cisco Fabric Manager Online Help*
- *Cisco Fabric Manager Web Services Online Help*

## Command-Line Interface

- *Cisco MDS 9000 Family Command Reference*

## Intelligent Storage Networking Services Configuration Guides

- *Cisco MDS 9000 I/O Acceleration Configuration Guide*
- *Cisco MDS 9000 Family SANTap Deployment Guide*
- *Cisco MDS 9000 Family Data Mobility Manager Configuration Guide*
- *Cisco MDS 9000 Family Storage Media Encryption Configuration Guide*
- *Cisco MDS 9000 Family Secure Erase Configuration Guide*
- *Cisco MDS 9000 Family Cookbook for Cisco MDS SAN-OS*

## Troubleshooting and Reference

- *Cisco NX-OS System Messages Reference*
- *Cisco MDS 9000 Family NX-OS Troubleshooting Guide*
- *Cisco MDS 9000 Family NX-OS MIB Quick Reference*
- *Cisco MDS 9000 Family NX-OS SMI-S Programming Reference*
- *Cisco MDS 9000 Family Fabric Manager Server Database Schema*

## 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.

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*



REVIEW-DRAFT - CISCO CONFIDENTIAL

# CHAPTER 1

## Web Services Overview

---

Cisco Fabric Manager Web Services provide application programming interfaces (APIs) that expose Fabric Manager core software functionalities as remote procedure calls to third-party vendors. Software developers can use the APIs to design computer applications that interact with the Fabric Manager Server over the network.

Using Fabric Manager you can monitor MDS switch events, performance, and inventory, and you can perform administrative tasks. Applications can access Fabric Manager Web Services through many protocols and data formats such as HTTP, HTTPS, XML, SOAP and WSDL.

Fabric Manager Web Services provide cross-platform operations. Web Services can interact with .NET applications, C++ applications, and applications written in other programming languages, and Web Services must adhere to accepted conventions to make services interoperable with other applications. For this reason, Fabric Manager Web Services must follow the Java API for XML Web Services (JAX-WS) specification. Fabric Manager Web Services relies on JBoss Web Service (JBoss WS) as a service endpoint engine and as an entry point into the JAX-WS programming model. The framework also allows Fabric Manager Web Services to become an integral part of the Fabric Manager Server run-time environment.

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*





## CHAPTER 2

# Fabric Manager Web Services

---

This chapter describes the Fabric Manager Web Services (FMWS) application program interface (API). This chapter includes the following sections:

- [About Fabric Manager Web Services, page 2-1](#)
- [Web Services Specifications, page 2-1](#)
- [Logon Service, page 2-2](#)
- [San Service, page 2-4](#)
- [Service Endpoint Interface \(SEI\), page 2-4](#)
- [Methods, page 2-4](#)
- [Error Codes, page 2-37](#)

## About Fabric Manager Web Services

The Cisco Fabric Manager Web Services (FMWS) enables third-party vendors to access Fabric Manager core software functionalities as remote procedure calls. Web Services extend the World Wide Web infrastructure to provide a method to softwares connecting to other software applications. Applications access Web Services using many protocols and data formats such as HTTP, HTTPS, XML, and SOAP. Web Services combine the best aspects of component-based development and the web. Web Services makes Fabric Manager an enterprise class application allowing it to be interoperable with other software platforms.

This chapter defines the APIs exposed by the Fabric Manager Web Services feature.

## Web Services Specifications

Web Services specifications combine together to provide interoperable protocols for security, communication and syntax for representing data.

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

## XML

XML is the data format that defines the structure of the message. XML Web Services architecture allows programs written in different languages on different platforms to communicate with each other in a standards-based way. XML Web Services expose useful functionality to Web users through a standard Web protocol (SOAP).

## SOAP

Simple Object Access Protocol (SOAP) is the communications protocol for Web Services. SOAP is a specification that defines the XML format for messages. The advantage of SOAP is that it has been implemented on many different hardware and software platforms.

## HTTP/HTTPS

HTTP/HTTPS is the transport layer of the service. HTTP/HTTPS allows data to traverse the network easily and is widely accepted. It is also considered as platform neutral. Every Fabric Manager Web Services operation is through HTTP/HTTPS.

## WSDL

A WSDL definition is an XML document with a root definition element from the <http://schemas.xmlsoap.org/wsdl/> namespace. Fabric Manager Web Services uses the WSDL document to publish which operations of Fabric Manager are available. The definitions element may contain several other elements including types, message, portType, binding, and service, all of which come from the namespace. WSDL is published on FMServer at <http://localhost/LogonWSService/LogonWS?wsdl>

# Logon Service

LogonWS makes IdentityManager's operations available as Web Service calls. LogonWS allows the following operations:

## requestToken

This method returns a token string that must be passed in as the header of the SOAP message. Once the username and password is authentication using Fabric Manager's SecurityManager, the token is generated and is kept valid for the number of milliseconds specified in the expiration argument.

### Parameters

username—Name of the user.

password—Password of the user.

expiration—Time (in milliseconds).

### Return Value

Session token.

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

#### Error

Error code: 201—Invalid argument in Web Service exception.

## validateToken

This method returns true or false depending on the validity of the token. If the token has expired, it returns false, or else it returns true.

#### Parameters

token—Session Token.

#### Return Value

Boolean value “True“ if the Fabric Manager accepts the token.

#### Error

Error code: 201—Invalid argument in Web Service exception.

## Authentication or Token

To interact with Fabric Manager Web Services, the user must obtain a token through LogonWS and attach this token to the header message of every SOAP requests. Fabric Manager Web Services verifies user credentials using a unique token string that is administered by LogonWS. At any given time, HTTPS should be deployed to secure the communication channel. The following example displays the format of the header message:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Header xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
    <m:Token xmlns:m="http://www.w3schools.com/transaction/">
      token string is put here
    </m:Token></SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <getFabrics xmlns="http://ep.jaxws.dcbu.cisco.com/" />
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

## IdentityManager

IdentityManager provides identity services and manages the user credentials that are required by Web Services. It is the token provider that administers and maintains tokens. It authenticates the user, generates tokens, and validates or expires tokens by periodically checking and clearing the cache.

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

## San Service

San Service is an Enterprise Java Beans (EJB) component that manages SAN-related service requests and executes queries on Fabric Manager for information. SanWS checks with IdentityManager for authentication before performing the request. A valid token string tells San Service that the user is a Fabric Manager user and it must honor and execute the request. After retrieving the required information it sends the result back to the user. SanWS logs errors in `fms_ws.log`.

## Service Endpoint Interface (SEI)

Service endpoint interface (SEI) of SanWS defines the operations of the service. These methods are published to the end users.

## Methods

### getFabrics

Returns the list of all open fabrics.

#### Return Value

An array of open fabrics.

#### Error

Error Code: 300— General SAN Service exception.

### getFabricByIP

Returns the list of fabrics associated with the IP address of a given switch.

#### Parameters

`ipAddress`—IP address of the switch.

#### Return Value

List of all fabrics associated with the specificIP address.

#### Error

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

### getFabricByKey

Returns the list of fabrics associated with the specified key.

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

**Parameters**

key—Key of the fabric.

**Return Value**

List of all fabrics associated with the specified key.

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getFabricBySwitchKey

Returns the list of fabrics associated with the specified seed switch key (WWN).

**Parameters**

swkey—Seed switch key of the fabric.

**Return Value**

List of all fabrics associated with the specified seed switch key.

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getSwitchesByFabric

Returns the list of switches associated with the specified fabric key.

**Parameters**

key—Key of the fabric.

**Return Value**

List of all fabrics associated with the specified fabric key.

**Error**

Error Code: 300— General SAN Service exception.

## getSwitch

Returns the list of switches on all the fabrics.

**Parameters**

key—Key of the fabric.

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

**Return Value**

List of all fabrics associated with the specified fabric key.

**Error**

Error Code: 300— General SAN Service exception.

## getSwitchByKey

Returns the switch associated with the specified switch key object.

**Parameters**

key—Key of the fabric.

**Return Value**

Switch associated with the specified switch key.

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getSwitchIPByName

Returns the IP address associated with the specified system name or switch name.

**Parameters**

sysname—Name of the system or switch.

**Return Value**

IP address associated with the specified system name.

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getSwitchIPByKey

Returns the IP address of the switch associated with the specified WwnKey object.

**Parameters**

key—WWN Key object.

**Return Value**

IP address associated with the specified WwnKey object.

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getNeighborSwitches

Returns the list of neighboring switches associated with the specified WwnKey.

**Parameters**

key—Wwn Key object.

**Return Value**

List of neighboring switches associated with the specifiedWwnKey.

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

Error code: 302—SAN does not found objects by query key exception.

## getVsans

Returns the list of VSANs in the fabric associated with the specified fabric key.

**Parameters**

key—fabric key object.

**Return Value**

List of VSANs in the fabric associated with the specified fabric key.

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getVsan

Returns the VSAN in the fabric associated with the specified VSAN key object.

**Parameters**

key—VSAN key object.

**Return Value**

VSANs in the fabric associated with the specified VSAN key object.

**Error**

Error Code: 300— General SAN Service exception.

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

Error code: 201—Invalid argument in Web Service exception.

## getIsIs

Returns the list of ISLs in the VSAN associated with the specified VSAN key.

### Parameters

key—VSAN key.

### Return Value

Array of ISL objects in the VSAN associated with the specified VSAN key.

### Error

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## discoverFabric

This API will open the fabric. This function requires the IP address of the seed switch and SNMP credentials.

### Parameters

seed—IP address of the seed switch.

user—SNMP Credential.

### Return Value

Boolean value is True, if the discovery was successful.

### Error

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

Error Code: 100— Authentication failure exception.

Error code: 101—Invalid credentials exception.

## manageFabric

Returns true or false depending managability of the fabric.

### Parameters

key—Fabric key.

### Return Value

Returns true if the fabric can be identified ot managed. Returns false if the fabric cannot be identified ot managed.



*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## unManageFabric

This function is used to unmanage a fabric.

**Parameters**

key—Fabric key.

**Return Value**

None.

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## closeFabric

This functions is used to unmanage and close a fabric.

**Parameters**

key—Fabric key.

**Return Value**

None.

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## purgeFabric

This functions is used to purge the specified fabric data both from Fabric Manager cache and database.

**Parameters**

key—Fabric key.

**Return Value**

None.

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

Error code: 302—SAN does not found objects by query key exception.

## getEndpoints

Returns the list of all the end ports known to the Fabric Manager.

### Return Value

An array of all the end ports.

### Error

Error Code: 300— General SAN Service exception.

## getEnclosures

Returns the list of all the enclosures known to the Fabric Manager.

### Return Value

An array of enclosure objects.

### Error

Error Code: 300— General SAN Service exception.

## getEndPointByKey

Returns the end port based on the Switch WWN.

### Parameters

key—WWN of the node.

### Return Value

Returns the end port based on the Switch WWN. Returns null if there are no end ports associated with the Switch.

### Error

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getEndPointAttachedToSw

Returns the end ports that are associated with a switch.

### Parameters

key—IP address of the switch.

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

**Return Value**

Returns the end ports based on switch.

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getEnclosureByName

Returns the enclosure based on the name.

**Parameters**

name—Name of the enclosure object.

**Return Value**

Returns the enclosure object..

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getEnclosureByKey

Returns the enclosure based on the name.

**Parameters**

name—Name of the enclosure object.

**Return Value**

Returns the enclosure object.

**Error**

Error Code: 300— General SAN Service exception .

Error code: 201—Invalid argument in Web Service exception.

## getEnclosureByPWWn

Returns the enclosure that are associated with a physical WWN.

**Parameters**

wwn—Physical WWN of the switch.

**Return Value**

Returns the enclosure based on physical WWN.

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## updateEnclosure

Update the enclosure with the value that is passed as parameter.

**Parameters**

value—Value to update the enclosure.

**Return Value**

None.

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## updateEndportEnclosure

Update the end port enclosure with the value that is passed as parameter.

**Parameters**

endportKey—Value for the endportKey.

enclosureKey—Value for the enclosureKey.

**Return Value**

None.

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getHosts

Returns the list of all the host enclosures known to Fabric Manager.

**Return Value**

Returns the list of all the host enclosures known to Fabric Manager.

**Error**

Error Code: 300— General SAN Service exception.

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

## getHost

Returns the name of hosts in a VSAN.

### Parameters

key—Name of the VSAN.

### Return Value

Returns the name of the hosts in the specified VSAN.

### Error

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getHostByFabric

Returns the name of hosts in a fabric.

ValidationException is thrown if any of the following situation occurs:

- If the argument passed is null.
- If the argument does not contain a valid key.

### Parameters

key—Name of the fabric.

### Return Value

Returns the name of the hosts in the specified VSAN.

### Error

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getStorages

Returns the list of all the storage device enclosures known to Fabric Manager.

### Return Value

An array of all the storage device enclosures known to Fabric Manager.

### Error

Error Code: 300— General SAN Service exception.

## getStorageByFabric

Returns the name of storage device enclosures in a fabric.

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

**Parameters**

key—Name of the fabric.

**Return Value**

Returns the name of the storages in the specified fabric.

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getHostPorts

Returns the list of all the host end ports in a fabric.

**Parameters**

key—Name of the fabric.

**Return Value**

An array of all the host ports in a fabric.

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getDomainId

Returns the domain address.

**Parameters**

key—Wwn

vsanid—Unique identifier of the VSAN

**Return Value**

Domain IP Address

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getVsanIp

Returns the IP Address of a VSAN

**Parameters**

key—Wwn

vsanid—Unique identifier of the VSAN

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

**Return Value**

IP Address of a VSAN

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getVsanDomains

Returns all VSAN domains in a switch.

**Parameters**

Key—Wwnkey

**Return Value**

VSAN domains in a switch.

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getIvrEnfZoneSetName

Returns the fabric IVR enforced zone set name.

**Parameters**

Key—Fabric key

**Return Value**

Zone set name.

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getIvrEnfZoneSetNumber

Returns the fabric IVR enforced zone number.

**Parameters**

Key—Fabric key

**Return Value**

Zone number.

**Error**

Error Code: 300— General SAN Service exception.

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

Error code: 201—Invalid argument in Web Service exception.

## getIvrEnfZoneSetActiveTime

Returns the fabric IVR enforced zone set activate time.

### Parameters

Key—Fabric key

### Return Value

Time stamp in the long integer format.

### Error

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getIvrEnfZoneSet

Returns the fabric IVR enforced zone set.

### Parameters

Key—Fabric key

### Return Value

List of zone objects.

### Error

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getIvrActiveZonesetChecksum

Returns the IVR active zoneset checksum.

### Parameters

Key—Fabric key

### Return Value

Checksum value.

### Error

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.



*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

## getAliases

Returns all the aliases used by the fabric.

### Parameters

Key—Fabric key

### Return Value

Aliases used by the fabric.

### Error

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## useFcAlias

Returns all the FC aliases used by the fabric.

### Parameters

Key—Fabric key

### Return Value

FC aliases used by the fabric.

### Error

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getEnfZoneSet

Returns all the VSAN enforced zone set.

### Parameters

Key—Fabric key

### Return Value

List of zones.

### Error

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getEnfZoneSetName

Returns all the VSAN enforced zone set name.

### Parameters

Key—Fabric key

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

**Return Value**

List of zone set names.

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getEnfZoneSetName

Returns all the VSAN enforced zone set name.

**Parameters**

Key—Fabric key

**Return Value**

List of zone set names.

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getFCAliases

Returns all the FC aliases for the fabric.

**Parameters**

Key—Fabric key

**Return Value**

List of aliases.

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getFCAliasesByVsan

Returns all the FC aliases for the VSAN.

**Parameters**

Key—Fabric key

**Return Value**

List of aliases for the VSAN.

**Error**

Error Code: 300— General SAN Service exception.

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

Error code: 201—Invalid argument in Web Service exception.

## getCFS

Returns CFS.

### Parameters

Key—Fabric key

### Return Value

List of CFS features.

### Error

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getCFSBySwitch

Returns CFS.

### Parameters

Key—Switch key

### Return Value

List of CFS features.

### Error

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getFcipProfiles

Returns FCIP profiles.

### Parameters

Key—Fabric key

### Return Value

List of FCIP profiles.

### Error

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getFcipProfilesBySwitch

Returns FCIP profiles based on a switch.

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

**Parameters**

Key—Switch key

**Return Value**

List of FCIP profiles.

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getFcipTunnels

Returns FCIP tunnels.

**Parameters**

Key—Fabric key

**Return Value**

List of FCIP tunnels.

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getFcipTunnelsBySwitch

Returns FCIP tunnels based on a switch.

**Parameters**

Key—Switch key

**Return Value**

List of FCIP tunnels.

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getFcipTunnelErrors

Returns FCIP tunnels errors.

**Parameters**

Key—Fabric key

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

**Return Value**

List of FCIP tunnel errors.

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getFcipTunnelErrorsBySwitch

Returns FCIP tunnels errors based on a switch.

**Parameters**

Key—Switch key

**Return Value**

List of FCIP tunnels errors.

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getFcipTunnelErrorsBySwitch

Returns FCIP tunnels errors based on a switch.

**Parameters**

Key—Switch key

**Return Value**

List of FCIP tunnels errors.

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getZoneMode

Returns zone operation modes.

**Parameters**

Key—Fabric key

**Return Value**

List of zone operation modes.

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

## getZoneModeByVsan

Returns zone operation modes for VSAN.

### Parameters

Key—VSAN key

### Return Value

List of zone operation modes.

### Error

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getZoneAttributes

Returns zone attributes.

### Parameters

Key—Fabric key

### Return Value

Zone attributes.

### Error

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getZoneAttributesByVsan

Returns zone attributes for VSAN.

### Parameters

Key—VSAN key

### Return Value

Zone attributes for VSAN.

### Error

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getSwitchPorts

Returns ports for the switch.

### Parameters

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

Key—Fabric key

**Return Value**

List of ports for a given switch.

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## isIVREnabled

Returns a boolean value depending on whether the IVR is enabled on the switch or not.

**Parameters**

Key—Switch key

**Return Value**

Boolean value.

**Error**

Error Code: 300— General SAN Service exception.

Error code: 201—Invalid argument in Web Service exception.

## getSwitchDateAndTime

Returns the switch time and date.

**Parameters**

Key—Switch key

**Return Value**

Boolean value.

**Error**

Error Code: 400—SnmpException.

## Cluster WS - SEI

Service endpoint interface (SEI) of Cluster WS defines the operations of the service. These methods are published to the end users.

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

## Method

### getSwitchesByFabricKey

Returns all the switches related to a fabric key.

**Parameters**

key—Name of the fabric.

**Return Value**

All the switches associated with the fabric.

**Error**

Error Code: 400—SnmpException

### getServerIpByFabricKey

Returns the managing server IP address from a fabric key.

**Parameters**

key—Name of the fabric.

**Return Value**

IP Address of the server.

**Error**

Error Code: 400—SnmpException

### getServerIpBySwitchKey

Returns the managing server IP address from a switch key.

**Parameters**

key—Name of the switch.

**Return Value**

IP Address of the Server.

**Error**

Error Code: 400—SnmpException

### getFabricsByServerIp

Returns name of the fabric by server IP address.

**Parameters**

key—IP Address.



*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

**Return Value**

Name of the fabric.

**Error**

Error Code: 400—SnmpException

## getAllServers

Returns all the servers in this federation

**Parameters**

key—

**Return Value**

All the servers in the federation.

**Error**

Error Code: 400—SnmpException

## getFabricByEnclosureKey

Returns the fabric key from an enclosure key

**Parameters**

key—Name of the enclosure.

**Return Value**

Name of the fabric.

**Error**

Error Code: 400—SnmpException

## getServerIpByEnclosureKey

Returns the server IP address from an enclosure key

**Parameters**

key—Name of the fabric.

**Return Value**

IP Address of the server.

**Error**

Error Code: 400—SnmpException

## getServerIpByVsanKey

Returns the server IP address from a VSAN key

**Parameters**

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

key—Name of the VSAN

**Return Value**

IP Address of the server.

**Error**

Error Code: 400—SnmpException

## Event WS - SEI

Service endpoint interface (SEI) of Event WS defines the operations of the service. These methods are published to the end users.

## Method

### isCallHomeEnabled

Returns a boolean value depending upon the activation of callhome feature.

**Parameters**

key—WwnKey

**Return Value**

Boolean value

**Error**

Error Code: 400—SnmpException

### getCallHomeDestProfile

Returns the callhome destination profile.

**Parameters**

key—WwnKey

**Return Value**

**Error**

Error Code: 400—SnmpException

### getCallHomeSysInfo

Returns system information about the callhome feature.

**Parameters**

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

key—WwnKey

**Return Value**

System information about the callhome feature.

**Error**

Error Code: 400—SnmpException

## getEmailMaxEntries

Returns the maximum number of email address entries for callhome feature.

**Parameters**

key—WwnKey

**Return Value**

Number of email address entries for callhome feature

**Error**

Error Code: 400—SnmpException

## getEmailSetup

Returns the email setup details of callhome feature.

**Parameters**

key—WwnKey

**Return Value**

Email setup details of callhome feature.

**Error**

Error Code: 400—SnmpException

## getSyslogServers

Returns the list of syslog servers.

**Parameters**

key—WwnKey

**Return Value**

List of syslog servers.

**Error**

Error Code: 400—SnmpException

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

## getSyslogMessageControl

Returns a list of syslog message configuration.

### Parameters

key—WwnKey

### Return Value

List of syslog message configuration.

### Error

Error Code: 400—SnmpException

## getSyslogLoggingCfg

Returns syslog logging configuration.

### Parameters

key—WwnKey

### Return Value

Configuration information on syslog credentials.

### Error

Error Code: 400—SnmpException

# Protocol WS - SEI

## Method

### getNtpPeers

Returns NTP peer information.

### Parameters

key—WwnKey

### Return Value

NTP peer information.

### Error

Error Code: 400—SnmpException

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

## getNtpInfo

Returns NTP system information

### Parameters

key—WwnKey

### Return Value

NTP system information.

### Error

Error Code: 400—SnmpException

## getFspfConfig

Returns the FSPF protocol configuration.

### Parameters

key—WwnKey

### Return Value

Configuration settings of the FSPF protocol.

### Error

Error Code: 400—SnmpException

## queryInterfaceFspfConfig

Returns the FSPF configuration on the interface pertaining to the specified VSAN.

### Parameters

key—WwnKey

vsanid—Unique identifier of the VSAN

### Return Value

FSPF configuration settings on the interface of a specified VSAN

### Error

Error Code: 400—SnmpException

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

## Security WS - SEI

### Method

#### getAaaMaxServer

Returns a value for the maximum number of server entries in a server group of AAA configuration.

**Parameters**

key—WwnKey

**Return Value**

Maximum number of server entries in a server group of AAA configuration.

**Error**

Error Code: 400—SnmpException

#### getAaaMaxAppServer

Returns a value for the maximum number of server entries in the AAA configuration for an application type.

**Parameters**

key—WwnKey

**Return Value**

Maximum number of server entries in a server group of AAA configuration.

**Error**

Error Code: 400—SnmpException

#### isMSCHAPRequired

Returns a boolean value to indicate if MSCHAP authentication mechanism is required for authenticating a user.

**Parameters**

key—WwnKey

**Return Value**

Boolean value to indicate if MSCHAP authentication mechanism is required for authenticating a user..

**Error**

Error Code: 400—SnmpException

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

## getAaaSetup

Returns the AAA configuration.

### Parameters

key—WwnKey

### Return Value

AAA configuration.

### Error

Error Code: 400—SnmpException

## getAaaAppServerGroups

Returns the AAA server groups for a specific application type.

### Parameters

key—WwnKey

### Return Value

AAA server groups for a specific application type.

### Error

Error Code: 400—SnmpException

## getAaaServerGroups

Returns all the AAA server group entries (a server group consists of a number of AAA servers implementing the same AAA protocol).

### Parameters

key—WwnKey

### Return Value

AAA server group entries.

### Error

Error Code: 400—SnmpException

## getSnmpUsers

Returns information about SNMP users.

### Parameters

key—WwnKey

### Return Value

Information about SNMP users.

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

**Error**

Error Code: 400—SnmpException

## getIPACLProfiles

Returns all the IP ACL profiles.

**Parameters**

key—WwnKey

**Return Value**

All the IP ACL profiles.

**Error**

Error Code: 400—SnmpException

## getSSHConfig

Returns the SSH configuration information.

**Parameters**

key—WwnKey

**Return Value**

SSH configuration information.

**Error**

Error Code: 400—SnmpException

## getSSHEnabled

Returns a boolean value to indicate if the SSH is enabled.

**Parameters**

key—WwnKey

**Return Value**

Boolean value to indicate if the SSH is enabled

**Error**

Error Code: 400—SnmpException

## isTelnetEnabled

Returns a boolean value to indicate if the telnet is enabled.

**Parameters**

key—WwnKey



*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

**Return Value**

Boolean value to indicate if the telnet is enabled.

**Error**

Error Code: 400—SnmpException.

## getPkiRsaKeys

Returns the PKI RSA key-pair entries.

**Parameters**

key—WwnKey

**Return Value**

PKI RSA key-pair entries.

**Error**

Error Code: 400—SnmpException.

## getPkiTrustPointNames

Returns a list of PKI trustpoint names.

**Parameters**

key—WwnKey

**Return Value**

A list of PKI trustpoint names.

**Error**

Error Code: 400—SnmpException.

## getPkiTrustPointNames

Returns a list of PKI trustpoint names.

**Parameters**

key—WwnKey

**Return Value**

A list of PKI trustpoint names.

**Error**

Error Code: 400—SnmpException.

## getPkiCert

Returns certificate information of a PKI trustpoint.

**Parameters**

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

key—WwnKey

**Return Value**

Certificate information of a PKI trustpoint.

**Error**

Error Code: 400—SnmpException.

## getPkiAction

Returns the PKI support action of a trustpoint.

**Parameters**

key—WwnKey

**Return Value**

PKI support action of a trustpoint.

**Error**

Error Code: 400—SnmpException.

## getPkiTrustPoint

Returns the PKI trust point information which consists the key-pair name, a list revocation methods and the contact HTTP URL of the external OCSP server for certificate revocation.

**Parameters**

key—WwnKey

**Return Value**

PKI trust point information.

**Error**

Error Code: 400—SnmpException.

## getFeatureControls

Returns all of the feature control names and their respective statuses.

**Parameters**

key—WwnKey

**Return Value**

Feature control names and statuses.

**Error**

Error Code: 400—SnmpException.

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

## getIkeFailRecoveryCfg

Returns the IKE configuration.

### Parameters

key—WwnKey

### Return Value

IKE configuration.

### Error

Error Code: 400—SnmpException

## getIkeCfgPolicies

Returns the policy that is used to set up the IKE tunnels.

### Parameters

key—WwnKey

### Return Value

Policy that is used to set up the IKE tunnels.

### Error

Error Code: 400—SnmpException.

## getIkeCfgInitiators

Returns the IKE initiator configuration information.

### Parameters

key—WwnKey

### Return Value

IKE initiator configuration information.

### Error

Error Code: 400—SnmpException.

## getIkeTunnels

Returns the IKE tunnels information.

### Parameters

key—WwnKey

### Return Value

IKE tunnels information.

### Error

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

Error Code: 400—SnmpException.

## getIPsecGlobalCfg

Returns the IPsec tunnel configuration information.

### Parameters

key—WwnKey

### Return Value

IPsec tunnel configuration information.

### Error

Error Code: 400—SnmpException.

## getIPsecXformSets

Returns the IPsec transform set information.

### Parameters

key—WwnKey

### Return Value

IPsec transform set information.

### Error

Error Code: 400—SnmpException.

## getIPsecCryptoMaps

Returns the IPsec cryptomap set.

### Parameters

key—WwnKey

### Return Value

IPsec cryptomap set.

### Error

Error Code: 400—SnmpException.

## getIfsFromCryptoMap

Returns the interface name from the IPsec cryptomap.

### Parameters

key—WwnKey

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

**Return Value**

Interface name from the IPsec cryptomap.

**Error**

Error Code: 400—SnmpException.

## getIPsecTunnels

Returns the information about IPsec tunnels.

**Parameters**

key—WwnKey

**Return Value**

Information about IPsec tunnels.

**Error**

Error Code: 400—SnmpException.

## Error Codes

Error Code	Description
100	Authentication failure.
101	Invalid credential.
102	Invalid privilege.
103	Invalid token.
200	Web Service error.
201	Invalid argument in Web Service function.
202	Unreachable Web Service server.
300	SAN service error.
301	Invalid query key
400	Snmp Error
201	InvalidArgument

**Note**

Fabric Manager Web Services supports server federation. Service requests to SanWS, SecurityWS, ProtocolWS, EventWS, and InventoryWS automatically dispatches the calls to correct server in the federation. If you are using server federation, the following methods will not automatically mediate to the corresponding server in the federation.

**SanWS:**

```
getEnclosures()
getEndports()
getFabricByIP()
getHosts()
```

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

```
getStorages()
getSwitchIPByName()
getSwitches()
```

**InventoryWS:**

```
getAllHbas()
getLicenseFlags()
```

In those specific instances, you may need to depend on ClusterWS to determine the server that you need to send the request.

---



## I N D E X

---

### C

#### CAs

description [2-1 to ??](#)

#### Cisco SAN-OS features

changed (table) [1-v](#)

new (table) [1-v](#)

closeFabric [2-9](#)

---

### D

#### digital certificates

description [2-1 to ??](#)

discoverFabric [2-8](#)

#### documentation

additional publications [i-viii](#)

related documents [i-viii](#)

---

### E

Error Codes [2-29](#)

---

### F

#### Fabric Manager features

changed (table) [1-v](#)

new (table) [1-v](#)

---

### G

getEnclosureByKey [2-11](#)

getEnclosureByName [2-11](#)

getEnclosureByPWwn [2-11](#)

getEnclosures [2-10](#)

getEndPortAttachedToSw [2-10](#)

getEndPortByKey [2-10](#)

getEndpoints [2-10](#)

getFabricByIP [2-4](#)

getFabricByKey [2-4](#)

getFabricBySwitchKey [2-5](#)

getFabrics [2-4](#)

getHost [2-13](#)

getHostByFabric [2-13](#)

getHostPorts [2-14](#)

getHosts [2-12](#)

getIsIs [2-8](#)

getNeighborSwitches [2-7](#)

getStorageByFabric [2-13](#)

getStorages [2-13](#)

getSwitch [2-5](#)

getSwitchByKey [2-6](#)

getSwitchesByFabric [2-5](#)

getSwitchIPByKey [2-6](#)

getSwitchIPByName [2-6](#)

getVsan [2-7](#)

getVsans [2-7](#)

---

### H

HTTP [2-2](#)

HTTPS [2-2](#)

---

### L

Logon Services [2-2](#)

*Send documentation comments to [mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)*

---

M

manageFabric [2-8](#)

---

P

purgeFabric [2-9](#)

---

S

Service Endpoint Interface [2-4](#)

SOAP [2-2](#)

---

U

unManageFabric [2-9](#)

updateEnclosure [2-12](#)

updateEndpointEnclosure [2-12](#)

---

W

WDSL [2-2](#)

---

X

XML [2-2](#)