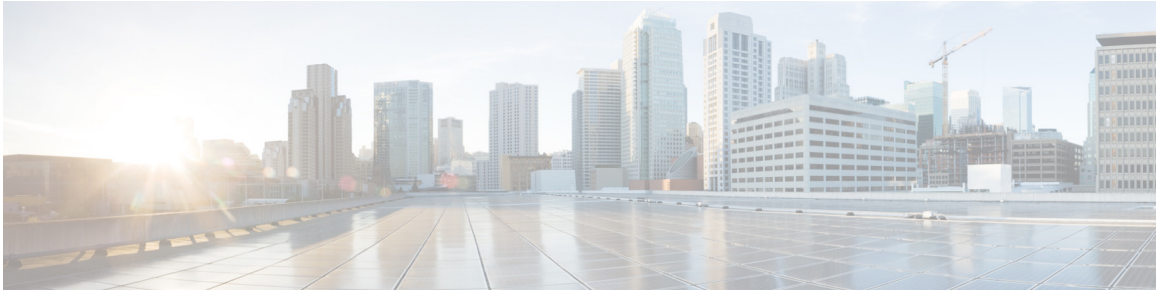
Cisco Configuration Engine Administration Guide 3.5.4

Cisco Systems, Inc.
www.cisco.com

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco website at www.cisco.com/go/offices.

Last Revised: April 10, 2014
Text Part Number: OL-17658-04

Cisco Configuration Engine Administration Guide 3.5.4



Preface

This preface describes the audience and conventions of the *Cisco Configuration Engine Administration Guide*. It also describes the available product documentation and provides information on how to obtain documentation and technical assistance. It contains the following sections:

- Audience
- Conventions
- Related Documentation
- Obtaining Documentation and Submitting a Service Request

Note This product contains cryptographic features and is subject to US and local laws governing import, export, transfer, and use.

Audience

This guide is intended primarily for:

- System administrators familiar with installing high-end networking equipment
- System administrators responsible for installing and configuring internetworking equipment who are familiar with Cisco IOS software

Conventions

This guide uses the following conventions:

Item	Convention
Commands and keywords.	**boldface** font
Variables for which you supply values.	*italic* font
Optional command keywords. You do not have to select any options.	[enclosed in brackets]
Required command keyword to be selected from a set of options. You must choose one option.	{options enclosed in braces | separated by vertical bar}
Displayed session and system information.	`screen` font
Information you enter.	**`boldface screen`** `font`
Variables you enter.	*`italic screen`* font
Menu items and button names.	**boldface** font
Choosing a menu item.	**Option > Network Preferences**

Note Means *reader take note*.

Tip Means *the following information will help you solve a problem*.

Caution Means *reader be careful*. In this situation, you might perform an action that could result in equipment damage or loss of data.

Timesaver Means *the described action saves time*. You can save time by performing the action described in the paragraph.

Warning **Means *reader be warned*. In this situation, you might perform an action that could result in bodily injury.**

Related Documentation

Table 1 describes the related documentation available for Cisco Configuration Engine.

Table 1 Cisco Configuration Engine Documentation

Document Title	Available Formats
Cisco Configuration Engine Installation and Configuration Guide 3.5.4	This document is available on *Cisco.com* and can be accessed without an account.
Cisco Configuration Engine Administration Guide 3.5.4	This document is available on *Cisco.com* and can be accessed without an account.
Cisco Configuration Engine Software Development Kit API Reference and Programmer Guide 3.5.3	This document is available on *Cisco.com* and can be accessed without an account.
Troubleshooting Guide for Cisco Configuration Engine 3.5.3	This document is available on *Cisco.com* and can be accessed without an account.
Cisco Configuration Engine 3.5.4 Open Source Documentation	This document is available on *Cisco.com* and can be accessed without an account.
Release Notes for Cisco Configuration Engine 3.5.4	This Release Note document is available on *Cisco.com* and can be accessed without an account.

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.

Product Overview

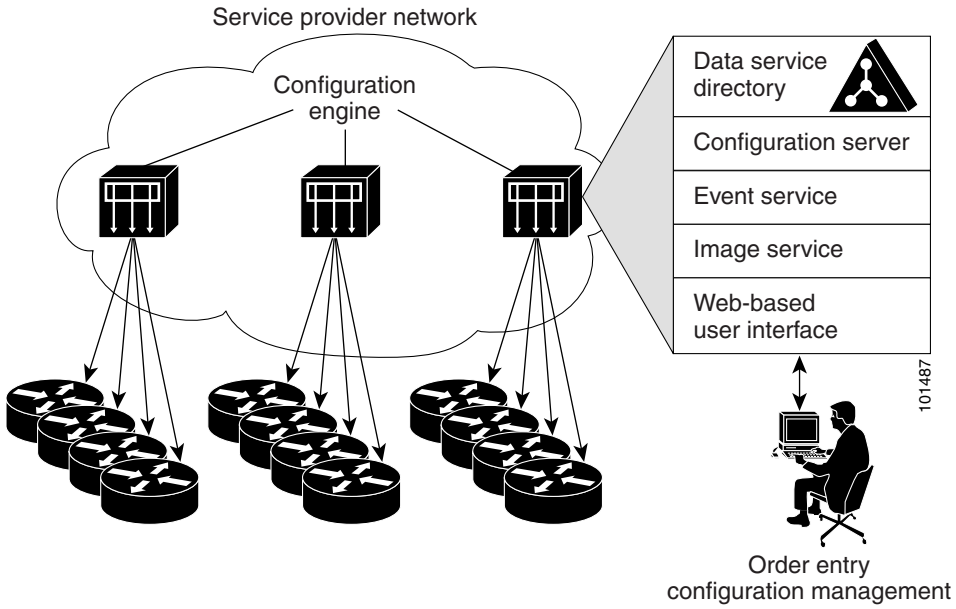
This chapter provides a high-level overview of the Cisco Configuration Engine 3.5.4. It is organized as follows:

- Cisco IOS Dependencies
- Modes of Operation
- Modes of User Authentication
- Configuration Service
- Event Service
- Dynamic Template and Object
- Image Service
- PIX Firewall Support
- ASA Firewall Support
- Intelligent Modular Gateway
- IMGW Device Module Toolkit
- Modular Router Support
- Encryption
- How the Cisco Configuration Engine Works
- Dynamic ConfigID and EventID Change Synchronization
- Common Log File Location

The Cisco Configuration Engine is a network management application that acts as a configuration service for automating the deployment and management of network devices and services (see Figure 1-1).

The Cisco Configuration Engine runs on Linux hardware platforms. For information about the supported hardware platforms, see *Cisco Configuration Engine Installation and Configuration Guide 3.5.4*.

Figure 1-1 Cisco Configuration Engine Architectural Overview



Each Cisco Configuration Engine manages a group of Cisco devices and services they deliver, storing their configurations and delivering them as needed. The Cisco Configuration Engine automates initial configurations and configuration updates by generating device-specific configuration changes, sends them to the device, executes the configuration change, and logs the results.

Note If you are running devices that use an earlier version of Cisco IOS, or a different operating system, such as Catalyst, you should invoke the Intelligent Modular Gateway for communicating with the device. For more information about Intelligent Modular Gateway, see "Intelligent Modular Gateway" section on page 1-10.

The Cisco Configuration Engine utilizes the following popular industry standards and technologies:

- eXtensible Markup Language (XML)
- Java naming directory interface (JNDI)
- Hypertext Transport Protocol (HTTP)
- Java servlets
- Lightweight Directory Access Protocol (LDAP)

The Cisco Configuration Engine supports two modes of operation (Internal Directory and External Directory) and it includes the following Cisco Configuration Engine components:

- Configuration service (web server, file manager, and namespace mapping server)
- Image Service (Cisco IOS images)
- Event service (event gateway)
- Data service directory (data models and schema)
- Intelligent Modular Gateway (IMGW)

The Cisco Configuration Engine can be used as the runtime component for deployment of customer-developed applications. These applications can be developed using the *Cisco Configuration Engine Software Development Kit API Reference and Programmer Guide*.

Supported Interfaces

The software external interfaces for Cisco Configuration Engine include:

- Linux login
- Telnet
- Secure Shell (SSH)

Cisco IOS Dependencies

Table 1-1 shows Cisco IOS versions with corresponding versions of Cisco Configuration Engine including feature limitations associated with each version.

Table 1-1 Cisco Configuration Engine 3.5.4 and Cisco IOS Dependencies

Cisco IOS	Cisco Configuration Engine	Limitations
12.3	1.3.2 or later	—
12.2(11)T	1.2 or later	—
12.2(2)T	1.2 or later with no authentication.	Applications will be unable to use exec commands or point-to-point messaging.

Third-Party Software

For information on the third-party software license, see the *Cisco Configurarion Engine 3.5.4 Open Source Documentation*.

Modes of Operation

There are two modes of system operation for the Cisco Configuration Engine:

- Internal Directory Mode
- External Directory Mode

Modes of User Authentication

There are two modes of user authentication for the Cisco Configuration Engine:

- Authenticate user internally
- Authenticate user externally

The Cisco Configuration Engine user can be authenticated internally or externally. The Cisco Configuration Engine user who logs in is authenticated against the external application. If the external authentication fails, the user is authenticated internally against the Cisco Configuration Engine LDAP Server. For more information about user authentication, see User Authentication, page 1-20.

Directory

Cisco Configuration Engine uses OpenLDAP for Directory services.

OpenLDAP can be configured to use internal or external database as data repository for the Directory. When configured to use external database (External Directory Mode), OpenLDAP stores data in relational tables using ODBC library.

OpenLDAP can also be configured to act as a proxy to forward incoming LDAP requests to another external LDAP server, which provides another possibility for string data in external LDAP server, for example, iPlanet.

Note GUI access to User Manager and Directory Manager is not available when operating in External Directory mode.

Configuration Service

The Configuration Service is the core component of the Cisco Configuration Engine. It consists of a configuration server that works in conjunction with configuration agents located at each router. The Configuration Service delivers device and service configurations to Cisco IOS devices for initial configuration and mass reconfiguration by logical groups. Routers receive their initial configuration from the Configuration Service when they start up on the network the first time.

The Configuration Service uses Event Service to send events required to apply configuration changes and receive success and failure notifications.

The configuration server consists of a web server that uses configuration templates and the device-specific configuration information stored in the embedded (Internal Directory mode) or remote (External Directory mode) directory.

Configuration templates are text files containing static configuration information in the form of command-line interface (CLI) commands. In the templates, variables are specified using (LDAP) URLs that reference the device-specific configuration information stored in the directory.

The configuration template includes additional features that allow simple conditional control structures and modular sub-templates in the configuration template (see Chapter 12, “Templates.”).

The configuration server uses HTTP to communicate with the Configuration Agent running on the managed Cisco IOS device. The configuration server transfers data in XML format. The configuration agent in the router uses its own XML parser to interpret the configuration data and remove the XML tags from the received configuration.

The configuration agent can also perform a syntax check on received configuration files. The configuration agent can also publish events through the event gateway to indicate the success or failure of the syntax check.

Event Service

The Cisco Configuration Engine uses the Event Service for receipt and generation of events. The Event Agent resides on Cisco IOS devices and facilitates communication between routers and the Event Gateway on the Cisco Configuration Engine.

The Event Service is a highly-scalable publish and subscribe communication method. The Event Service uses subject-based addressing to help messages reach their destination. Subject-based addressing conventions define a simple, uniform namespace for messages and their destinations.

NameSpace Mapper

The Namespace Mapping Service (NSM) allows you to address multiple network devices by a single posting of a publish or subscribe event. NSM also allows your network administrator to map Cisco-standardized event names to names of your choice.

For example, in a network of 100 routers, there might be 10 that the administrator wants to configure as a VPN (Virtual Private Network). In order to load a configuration into each of these devices, your client application could either publish 10 *cisco.mgmt.cns.config.load.<deviceId>* events, or the administrator could associate the 10 devices with a common group name and your client application can post the event once. The associated administration steps are:

1. Using the device management interface, define all the device objects (see Chapter 3, "Device and Subdevice Manager").
2. Using NSM administration interface, remap both the subscribe and publish map of *cisco.mgmt.cns.mgmt.config.load* subject to *application.loa*d (see Chapter 7, "Namespace Manager").
3. For example, using the group management interface, group all the devices in the West Coast under a group called "westcoast" (see Chapter 6, "Groups").
4. The client application would publish the mapped subject *application.load./config/westcoast* on the event bus and the devices in the "westcoast" group would get the event. The mapped subject is returned to the client application by the NSM's operational API when querying for the publish mapping for the event *cisco.mgmt.cns.config.load*.

Event Gateway

The Event Gateway acts as a relay between the Integration Bus and agent-enabled devices, which enables event-based communication. The Event Gateway uses NSM to map subjects.

Each Event Gateway process can support a maximum of 500 devices. To support more than 500 devices, you should run multiple gateway processes.

During the **Setup**, you can set the number of concurrent gateway processes to start with either one or both of the following prompts, depending on how you want to setup your Secure Socket Layer (SSL) (see "Encryption" section on page 1-12) communications:

```
Enter number of Event Gateways that will be started with crypto operation:
Enter number of Event Gateways that will be started with plaintext operation:
```

Event Gateways that listen on port 11011 and 11012 are Dispatcher Event Gateways which redirect a device connection to a regular plain-text or crypto enabled Event Gateway respectively. For more information see Chapter 6 "Scalability among Event Gateways" in *Cisco Configuration Engine Installation and Configuration Guide*.

Event Gateway Port Automatic Assignment

Each event gateway can support a maximum of 500 devices. During Zero Touch Deployment (ZTD), the deployment engineer needs to update the bootstrap configuration file for every 500 devices. The event gateway port automatic assignment process helps to eliminate the manual process. When the Cisco Configuration Engine server is configured as the previous section, all the 30,000 devices can be deployed using the same bootstrap configuration file. The following is the sample bootstrap configuration file. The bolded lines are the required commands to support the port automatic assignment.

```
cns trusted-server all-agents ce-host
cns id hardware-serial
cns id hardware-serial event
cns config initial ce-host status http://ce-host/cns/PostStatus
cns event ce-host keepalive 120 1 reconnect 10
cns config partial ce-host
```

When a network element connects to Cisco Configuration Engine through the dispatcher event gateway, the Cisco Configuration Engine automatically assigns a port to the network element. The network element saves that information and connects to the designated Cisco Configuration Engine port. The Cisco Configuration Engine can manage a device after the device connects to a none-Cisco Configuration Engine well-known port (ports other than 11011 and 11012).

Dynamic Template and Object

The original servlet, *com.cisco.cns.config.Config*, gets the configuration template from the attribute value of the Device Object in the configuration server data store (LDAP server), parses the template, and does string substitution on parameters inside the template. It is tightly coupled with the template that is assigned to the device and the attributes of device object.

The new servlet, **DynaConfig**, loosens the restriction so that the template can be assigned dynamically and the parameter values can be obtained from other objects in data store.

This servlet gets **PathInfo** information by means of **HttpServletRequest.getPathInfo()**, parse it, and gets the related template name and object reference. The structure of **PathInfo** is:

/*<argument name>=<argument value>*.

Data Structures

The feature of dynamic template and object utilizes **PathInfo**, which is passed from the client side to the servlets. The structure of **PathInfo**, which the servlet can understand is in following format:

```
[/<argument name>=<value>]*
```

The argument and format for dynamic template and object is:

```
[/cfgtpl=value[/object=value]]
```

For more information about Dynamic Template and Object, see *Cisco Configuration Engine Software Development Kit API Reference and Programmer Guide* .

Image Service

The Image Service is an automated, scalable, and secure mechanism designed to distribute Cisco IOS images and related software updates to Cisco IOS devices that have Cisco Intelligence Agents.

All the image upgrading decisions are made by the image server. These decisions are based on the inventory response information returned by the image agent.

imageInventoryResponse Message

The **imageInventoryResponse** message contains an imageInventoryReport XML document. This report contains information about:

- The running image on the system
- The systems hardware resources
- The various file systems and files on the device.

The **imageInventoryResponse** is a response to an imageInventoryRequest. The resources requested by the tags in the request are sent in the **imageInventoryResponse** message. The messageID element from the request is included in the messageID element of the response message.

For the devices hardware resources, the minimum information reported is:

- Size of the system RAM available to run an image
- Name(s) of the system (hostname and, imageID)
- Type of the device hardware
- Serial numbers of various hardware components
- Currently running system image on the managed device provides the following information:
  - Image file name and location, for example *flash:/c2600-is-mz*
  - MD5 hash of image file if it can be calculated
  - Version string, for example *IOS (tm) C2600 Software (C2600-IS-M) Version 12.2(10.7)T, MAINTENANCE INTERIM SOFTWARE*
- The date and time that the image was booted
- In addition, for each local persistent file system on the device, the following information is reported:
  - Name of file system
  - Type of the file system
  - Size of file system
  - Free space available
  - Read/Write protect flags
- For each file in each of the reported file systems, the following information is reported:
  - Name (both file name, and the complete fully qualified path name)
  - Size
  - R/W permission flags
  - Modification date
- For each directory in the file system, the following information is reported:

- Name (both directory name and the complete fully qualified path name)
- R/W permission flags

Image Update Criteria

When Image Service is instructed to evaluate a given device for distribution and/or activation, an **ImageCheckServer** message is sent over the Event Bus to get Inventory and analyze the inventory content to decide what attributes should be used for the comparison.

Currently, the following values are used from Inventory to determine which comparison class to use:

- MD5
- ImageFile
- File System

Distribution Decision Keys

File System Activation decision keys:

- ImageFile
- MD5
- Version String

Image Service makes decisions in the following order:

1. If MD5 and File System exist:
   a. Distribution:
   - If **Destination** in Distribution object exists on File System in Inventory, it is not necessary to distribute this file if *Overwrite* flag is not set. For example, **Destination** is *slot0:pf-1.img4*, if inventory return by device has a file *pf-1.img4* on slot0, Server decides this distribution is not needed.
   - If **Destination** does not exist in File System in Inventory, it starts to check if there is enough space left for this file on that location.

     If **Erase** is checked, server gets total size of that file system (that is, slot0) to see if the file can fit into this file system. For example, if slot0 has 1000 bytes free, 2000 bytes total size, and file size on distribution is 100 bytes, server does 2000 - 100 to check if the result is >0. If >0, it is okay to distribute.

     If **Overwrite**, server gets remaining free space size of that file system and adds the original file size on Inventory back, then it sees if the file will fit into this file system. For example, if slot0 has 1000 bytes free, the file is 100 bytes on inventory, the file size on distribution is 200 bytes and **Overwrite** is set, server does 1000 + 100 - 200 to check if slot0 remaining free size is > 0. If >0, it is okay to distribute.

   b. Activation:

     Server uses MD5 to compare between **RunningImageInfo** from Inventory and **ImageObject** on server side. If they are the same, Activation is not necessary.
2. If ImageFile and File System exists:
   a. Distribution: (The same as 1a)
   b. Activation:

Server compares *ImageFile* in **RunningImageInfo** from Inventory with **Destination** attribute on Distribution Object on server side. If they are the same, Activation is not necessary.

3. If Version String and File System exists:
   a. Distribution: (The same as 1a)
   b. Activation:

      Server compares *Version String* in **RunningImageInfo** from Inventory with *Description* on Image Object from server side. If they are the same, Activation is not necessary.
4. If Only ImageFile exists:
   a. Distribution:

      Server always thinks Distribution is necessary. (Because server uses *ImageStatus* message to verify if the result of Distribution is successful.)
   b. Activation: (The same as 2b)
5. If Only Version String exists:
   a. Distribution: (The same as 4a)
   b. Activation: The same as 3b)
6. If Only File System exists:
   a. Distribution: (The same as 1a)
   b. Activation:

      Server always thinks Activation is not necessary. (Because there is no way to verify if the result of Activation is successful.)
7. If none of those attributes exists in Inventory:
   a. Distribution**:**

      Server always thinks Distribution is not necessary.
   b. Activation:

      Server will always think Activation is not necessary.

For more information about how to use the Image Service, see Chapter 18, "Image Service."

For those devices that do not have a Cisco image agent, non-Cisco IOS devices, and non-Cisco devices, you can use the IMGW Toolkit to create scripts that support SSH sessions between these devices and the Cisco Configuration Engine.

For more information about the IMGW Device Module Toolkit, see Chapter 23, "IMGW Device Module Development Toolkit."

PIX Firewall Support

Cisco Configuration Engine provides configuration management and image service to Cisco PIX firewall devices (PIX device).

For more information about PIX firewall support, see Chapter 21, "PIX Firewall Device Support."

ASA Firewall Support

Cisco Configuration Engine provides configuration management and image service to Cisco daptive Security Appliance devices (ASA device).

For more information about PIX firewall support, see Chapter 22, "ASA Firewall Device Support."

Intelligent Modular Gateway

Intelligent Modular Gateway (IMGW) allows you to run the Cisco Configuration Engine for automatically distributing configuration files to Cisco IOS network devices running Cisco IOS versions earlier than 12.2(2)T, as well as to Catalyst switches, CCS 11k devices, Cache Engines, and PIX firewalls.

Note If you are running devices that use Cisco IOS version 12.2(2)T or later, you should use the Event Gateway.

The IMGW accomplishes this task by adding the ability to use alternate access methods (Telnet and SSH) to connect to devices that do not have Cisco Configuration Engine agents in their software.

The interface to the IMGW is the same as that of the Event Gateway. It responds to the same events. The NSM operates in the same way. Therefore, after some initial setup work is done, applications need not know the difference between communicating with agent-enabled devices by way of the Event Gateway and non-agent devices by way of the IMGW.

Restrictions

Using the IMGW with an SSH transport creates some restrictions in terms of how the Cisco Configuration Engine architecture is used.

- When using SSH as a transport, no syntax checking can be done on the configurations before they are applied.

  Syntax checking in the Cisco Configuration Engine architecture is accomplished by an intelligent agent in the device that has access to internal parser functions. An SSH interface does not provide any means to access this functionality. Therefore, any syntax checking attributes are ignored. Errors are only detected when the configuration is actually applied and applications must deal with the fact that configuration lines prior to the error were executed.

- Because all logic is external to the device, there is no way to watch for configuration changes that are done outside the scope of the network management software.

  For example, if a network administrator uses a standard SSH client to directly access a network element and changes the configuration, that element would not be synchronized with the network management infrastructure, and depending on the change, might become unmanageable. This is especially true if the login mechanisms (usernames and passwords) are changed. Login mechanism changes should be handled during a maintenance window, during which event-based configuration is not occurring, so that race conditions do not occur. Any such changes must be reflected on the provisioning system's device information screen so that the Device Information Database is properly updated before any new partial configurations are sent.

- The scope of error checking upon configuration load is limited to syntax checking.

Semantic errors cannot be detected. The output is returned in a buffer that applications should log. In a case where something is not operating properly, a network administrator can manually look at the log of what the device was reporting and determine if a semantic error occurred.

- The initial configuration mechanism as defined in the Cisco Configuration Engine architecture is not supported.

  This mechanism allows a router to be preconfigured with the **cns config initial** command, causing it to contact the configuration server to retrieve its initial configuration. However, because the legacy devices do not have the agent code in them, they can never contact the configuration server (they do not understand the configuration command). Therefore, this mechanism does not make sense when using SSH as a transport. If an initial configuration needs to be delivered by the Cisco Configuration Engine, it has to be done through the partial configuration mechanism.

- Aside from the device information database, the gateway is stateless.

  There is no read back of configurations to make sure they were applied, nor is there automatic rollback of configurations if a failure occurs.

- If a device is not directly connected to the management network, it must be attached through a Cisco communication servers.

  The API allows you to set up an arbitrary network topology to reach the device. However, this release only supports two possible topologies: direct connection to one of the device network interfaces, or console access by way of a Cisco access server, such as a 2511.

- Device failures are only detected within a user-specified polling interval.

  This is because while the standard Event Gateway requires that routers maintain a connection to the Event Gateway (so any breakage of that connection would signal a problem), the SSH interface is implemented through a transient connection. Therefore, the gateway must poll all devices at some user-specified interval to make sure they are responding, so failure detection is not immediate.

- When both agent-enabled and legacy devices are present on the same network, it is recommended that both gateways be run at the same time.

  The standard Event Gateway talks to the agent-enabled devices and the Intelligent Modular Gateway talks to the legacy devices.

Note Do not put an entry in the Device Information Database for a router that is already agent-enabled because both gateways will try to control the router and unpredictable results might occur.

IMGW Device Module Toolkit

The IMGW Device Module Toolkit allows you to develop your own device modules, plug them into Cisco Configuration Engine, then use them to configure devices.

For more information about the IMGW Device Module Toolkit, see Chapter 23, "IMGW Device Module Development Toolkit."

Modular Router Support

Cisco Configuration Engine supports modular routers. A modular router chassis includes slots in which you can install line and network interface cards.

For a modular router, a subdevice configuration object and configuration template is defined for every network module whose interfaces need to be configured and for which the interface number can be variable; based on the slot. Then, a device configuration object and a template is defined for the main device. Fixed interface numbers can be configured in the main device template.

Modular router events are published to the event bus and are accessible to applications connected to the bus. The Cisco IOS device publishes the system hardware configuration in the *cisco.mgmt.cns.inventory.device-details* event after hardware discovery. The Cisco Configuration Engine is configured to listen for this event, retrieve it and extract the hardware configuration of the device.

Encryption

The SSL method has been adopted as the encryption mechanism for HTTP sessions between the configuration agent and the configuration server, and the TCP session between the Event Gateway and the event agent.

To use encryption, the Cisco IOS devices must be running a crypto image and version 12.2(11)T of the Cisco IOS.

Device Authentication

The configuration server and Event Gateway are supplied with a X.509 certificate generated by a certificate authority (CA) server. It is the responsibility of the network administrator to have a CA server and to control certificate generation and revocation.

To be configured, the Cisco IOS device must be recognized by the CA. There is no client-side certificate in the Cisco IOS device.

For the configuration server, after the Cisco IOS device has validated the certificate, it sends a password over the encrypted pipe. The device uses the password to be authenticated by the Cisco Configuration Engine.

Note Authentication is also done when the links are in clear text.

A server configured for secure connections is also able to enact non-secure (clear-text) sessions. The password check is done regardless of whether encryption is used or not.

After the server is secured, it is no longer be able to process requests that do not have a password. It cannot tell the difference between a clear text request from a device in a secure environment or from a device in an non-secure environment.

For the Event Gateway, after the Cisco IOS device has validated the certificate, it sends a DeviceID control message over the encrypted pipe that has the Cisco Configuration Engine password of the device. The **event_id:cns_password** is validated using the authentication API. If it is not matched, the SSL session is terminated and an entry is made to the security log. This ensures only authorized customer premises equipment (CPE) devices connect to the Event Gateway and are able to use the Integration Bus.

Bootstrap Password

Cisco Configuration Engine provides a bootstrap password for use where multiple devices are deployed in a batch. In this case, all devices in a particular batch are given the same (bootstrap) password to use when they each start up on the network for the first time.

The bootstrap password can be changed for different batches of devices by using the **BootStrap** function under Security Manager in the user interface (see Chapter 13, "Security Manager").

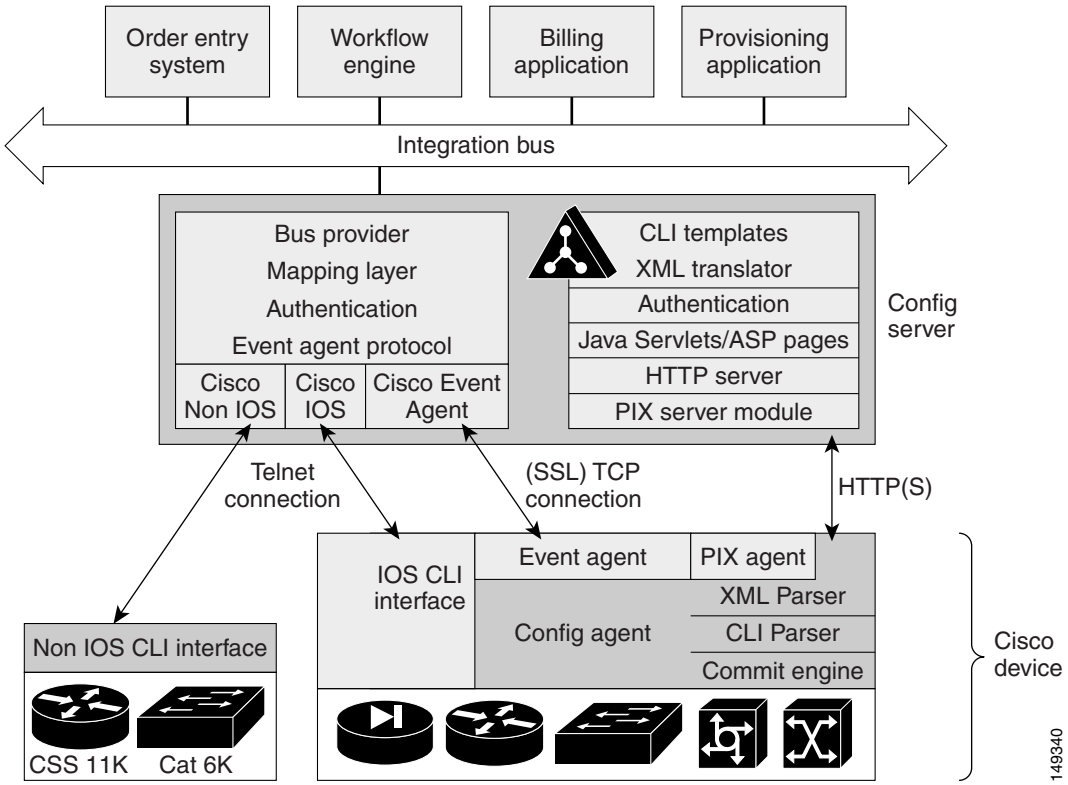
Resynchronize cns_password

If the password of a device becomes corrupted so that there is a mismatch between the device and the corresponding password information held in the Cisco Configuration Engine directory, you can resynchronize the device with the Cisco Configuration Engine by using the **Resync Device** function in the user interface (see "Resynchronizing Devices" section on page 3-62).

How the Cisco Configuration Engine Works

The Cisco Configuration Engine dynamically generates Cisco IOS configuration files (documents), packages these file in XML format, and distributes them by means of Web/HTTP (see Figure 1-2). This takes place in response to a *pull* (get) operation.

Figure 1-2 Configuration Engine Functional Diagram

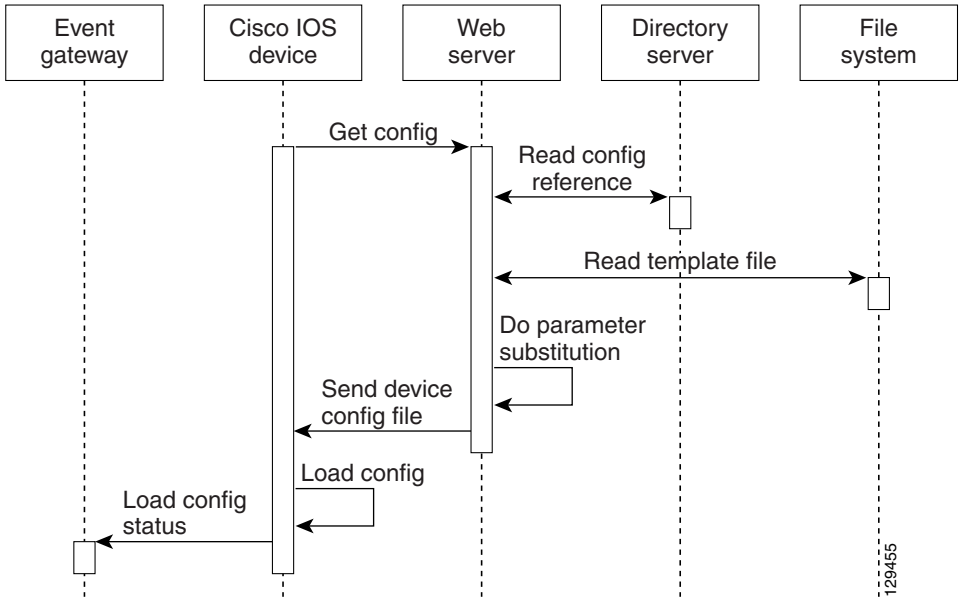


A Cisco IOS device initiates a get operation when it first appears on the network (**cns config init…**) or when notified (by subscribed event) of a configuration update (**cns config partial…**).

Note For more information about these and other related CLI commands, see the Cisco IOS configuration guide and command reference publications.

When a Cisco IOS device issues a request for a device configuration file, the request includes a unique identifier (configID = hostname) used to help locate the relevant configuration file parameters for this device on the directory server. Figure 1-3 shows the process flow for a configuration load operation.

Figure 1-3 Configuration Load Process Flow



When the web server receives a request for a configuration file, it invokes the Java Servlet and executes the embedded code. This directs the web server to access the directory server and file system to read the configuration reference for this device and template. The configuration server prepares an instantiated configuration file by substituting all the parameter values specified in the template with valid values for this device. The configuration server forwards the configuration file to the web server for transmission to the Cisco IOS device.

The configuration agent at the router accepts the configuration file from the web server, performs XML parsing, syntax checking (optional), and loads the configuration file. The router reports the status of the configuration load as an event that can be subscribed to by a network monitoring or workflow application.

Load Initial Configuration

1. The Cisco Configuration Engine reads the template files.
2. The Cisco Configuration Engine does the parameter substitution.
3. The Cisco Configuration Engine sends the device configuration to the Cisco IOS device.
4. The Cisco IOS device tries to load the initial configuration.

5. The Cisco IOS device publishes the load configuration status event to the event gateway.

Modular Router

1. The modular router posts an HTTP request containing the hardware configuration to the Cisco Configuration Engine for the initial configuration.
2. The Cisco Configuration Engine reads the hardware configuration of the device from the HTTP request and updates the directory server with the latest configuration.
3. The Cisco Configuration Engine reads the template files.
4. The Cisco Configuration Engine does the parameter substitution.
5. The Cisco Configuration Engine sends the device configuration to the Cisco IOS device.
6. The modular router tries to load the initial configuration.
7. The modular router publishes the load configuration status event to the event gateway.

Load Partial Configuration

1. The user modifies a template in the Cisco Configuration Engine user interface.
2. The template contents are passed to the Cisco Configuration Engine.
3. The Cisco Configuration Engine stores the template in the file system.
4. The user clicks the update device button in the user interface.
5. The Cisco Configuration Engine publishes a *cisco.mgmt.cns.config.load* event.
6. The Cisco IOS device receives the *cisco.mgmt.cns.config.load* event and in response to this event requests its configuration by contacting the server.
7. The Cisco Configuration Engine reads the template files.
8. The Cisco Configuration Engine sends the device configuration to the Cisco IOS device.
9. The Cisco IOS device tries to load the partial configuration.
10. The Cisco IOS device publishes the load configuration status event to the event gateway.

Modular Router

1. The user modifies a template in the Cisco Configuration Engine user interface.
2. The template contents are passed to the Cisco Configuration Engine.
3. The Cisco Configuration Engine stores the template in the file system.
4. The user clicks the update device button in the user interface.
5. The Cisco Configuration Engine publishes a *cisco.mgmt.cns.config.load* event.
6. The modular router retrieves the *cisco.mgmt.cns.config.load* event and in response to this event requests its configuration by contacting the server.
7. The Cisco IOS device posts a HTTP request containing the hardware configuration to the Cisco Configuration Engine for the partial configuration.
8. The Cisco Configuration Engine reads the hardware configuration of the device from the HTTP request and updates the directory server with the latest configuration.The Cisco Configuration Engine does the parameter substitution.
9. The Cisco Configuration Engine reads the template files.

10. The Cisco Configuration Engine does the parameter substitution.
11. The Cisco Configuration Engine sends the device configuration to the modular router.
12. The modular router tries to load the partial configuration.
13. The modular router publishes the load configuration status event to the event gateway.

EventIDs and ConfigIDs

The Cisco Configuration Engine intersects two name space domains:

- Configuration Domain
- Event Domain

The Cisco Configuration Engine uses the Configuration Domain when a device communicates with the configuration server. It uses the Event Domain when a device communicates with the Cisco Configuration Engine using the publish and subscribe mechanism of the Integration Bus.

The device must be uniquely identified in these namespaces. The ConfigID uniquely identifies the device in the Configuration Domain. The EventID uniquely identifies the device in the Event Domain.

Because the Cisco Configuration Engine uses both the Integration Bus (event bus) and the configuration server to provide configurations to devices, both EventID and ConfigID must be defined for each configured Cisco IOS device.

The values for EventID and ConfigID for each device can be identical, or you can make them different when you add or edit device information using the user interface (see "Editing Devices" section on page 3-53).

Dynamic ConfigID and EventID Change Synchronization

The Cisco IOS, version 12.2.(11)T, was enhanced with new CLI ID commands that can modify the EventID and ConfigID, then reconnect the device to the Cisco Configuration Engine with the new IDs.

Common Log File Location

In Cisco Configuration Engine, all log files go into */var/log/CNSCE/<modulename>*. For all Cisco Configuration Engine logs, this feature also includes custom logrotate scripts, located in the */etc/logrotate.d/cnsce* directory.

Logrotate is a system utility that can rotate specified log files according to the conditions specified in a config file. There is a config file defined for each module (see "Sample Logrotate Config File" section on page 1-17). An Administrator-level user can make use of these config files to rotate logs of any module at any time.

For example, the command **logrotate -f /etc/logrotate.d/cnsce/imgw** rotates all IMGW logs and backs up all existing logs in the */var/log/CNSCE_ROTATED_LOGS* directory. This is a common backup directory where all the rotated logs for all the modules are dumped.

Having a common directory allows you to set aside separate partition, or space, for backup logs.

Sample Logrotate Config File

```
#-----------------------------------------------------------------
# Copyright (c) 2002, 2003, 2004 by Cisco Systems, Inc.
# All rights reserved.
#-----------------------------------------------------------------
/var/log/CNSCE/imgw/* {
daily
missingok
copytruncate
compress
olddir /var/log/CNSCE_ROTATED_LOGS
}
```

Dynamic Log level Update

With this release, you can now change the log level programmatically using Web Services. A new API has been defined in Admin Web Service: *setLogLevel(int level, Token token)*.

```
    /**
    * Changes the logging level of CE components.
    *
    * @param level, the logging level. Allowed values debug, info, warn, error
    *, fatal
    * @param token a Token object.
    * @return int, the new Log level.
    * @throws AdminServiceException if there is an error setting the log level.
    * @throws RemoteException if there is an error communicating with the service.
    */
int setLogLevel(int level, Token token)
throws AdminServiceException, RemoteException ;
For debug, set level =1.
For info, set level =2.
For warn, set level =3.
For error, set level =4.
For fatal, set level =5.
```

Monitoring Service

A wrapper monitoring service is provided in this release to monitor the various Cisco Configuration Engine services. If any of the Cisco Configuration Engine processes die, the monitoring service exits.

Other applications can monitor this single Cisco Configuration Engine process, rather than all dependent Cisco Configuration Engine services. In the case of failure, they can take appropriate action, such as invoking the restart script.

Other applications can check for the existence of this wrapper monitoring process to make sure that all Cisco Configuration Engine services are up. If the process is not running, it will signify that one or more of Cisco Configuration Engine services are down.

This service report the health of the various Cisco Configuration Engine processes in a log file. If there is a failure, the service reports the error and exits. A time stamp is appended to each report.

There is a provision to start, stop, or check the status of this service. The following Cisco Configuration Engine processes are monitored:

- HTTP/Tomcat
- Event Gateway
- IMGW
- Web Services
- Tibco Rendezvous Daemon

Health Checking Utility

A wrapper resource health checking utility is provided in release 3.5 to monitor the health of the Event Gateways and Tibco Rendezvous Daemon. If any of the processes stops, the health checking utility restarts the process and logs a message in **/var/log/CNSCE/ce_resource/ce_resource.log** file. This utility (resource_monitor_daemon) starts during the Cisco Configuration Engine setup and stops when the Cisco Configuration Engine server stops.

Software Architecture

The monitoring service is a single process running as a daemon on the Cisco Configuration Engine host system. This daemon checks the state of various Cisco Configuration Engine processes at regular interval of time. This time interval is configurable. If any of the process in Cisco Configuration Engine dies, the daemon exits. A shell script is provided to start, stop, or check the status of this daemon. Applications can check the Cisco Configuration Engine health using this shell script.

Daemon Start/Stop script

The **MonitorCE** shell script starts and stops the daemon. This script also provides the status of the daemon script. Integrating applications use this shell script to monitor the state of Cisco Configuration Engine services.

This script is registered as a start up script on the local OS using the **chkconfig** utility. In this way, the script is started automatically after the host system is restarted. The script is located in the */etc/rc.d/init.d* directory.

Logging

The daemon checks for the health of each Cisco Configuration Engine process and reports it in a log file. The log files are located in */var/log/CNSCE/ce_health/ce_monitor.log*. A time stamp is appended with each report.

Here is an example of the log file:

```
07/14/2005-06:53 HTTP/Tomcat is UP in plain-text mode.
07/14/2005-06:53 HTTP/Tomcat is UP in ssl mode.
07/14/2005-06:53 Event Gateway (plaintext operation) at port 11011 is UP.
07/14/2005-06:53 Event Gateway (plaintext operation) at port 11013 is UP.
07/14/2005-06:53 Event Gateway (plaintext operation) at port 11015 is UP.
07/14/2005-06:53 Event Gateway (plaintext operation) at port 11017 is UP.
07/14/2005-06:53 Event Gateway (crypto operation) at port 11012 is UP.
```

```
07/14/2005-06:53 Event Gateway (crypto operation) at port 11014 is UP.
07/14/2005-06:53 Event Gateway (crypto operation) at port 11016 is UP.
07/14/2005-06:53 IMGW is UP.
07/14/2005-06:53 Cisco-CE Event Bus is UP.
07/14/2005-06:53 CEAdminService web service is UP in plain-text mode.
07/14/2005-06:53 CEConfigService web service is UP in plain-text mode.
07/14/2005-06:53 CEImageService web service is UP in plain-text mode.
07/14/2005-06:53 CEExecService web service is UP in plain-text mode.
```

When HTTP is Down

Here is an example when HTTP is down:

```
07/14/2005-06:53 HTTP/Tomcat is DOWN in plain-text mode.
HTTP GET failed on URL http://infystorm5:80/cns/Config
Connection refused

07/14/2005-06:53 HTTP/Tomcat is DOWN in ssl mode.
HTTP GET failed on URL https://infystorm5:444/cns/Config
Connection refused

07/14/2005-06:53 Event Gateway (plaintext operation) at port 11011 is UP.
07/14/2005-06:53 Event Gateway (plaintext operation) at port 11013 is UP.
07/14/2005-06:53 Event Gateway (plaintext operation) at port 11015 is UP.
07/14/2005-06:53 Event Gateway (plaintext operation) at port 11017 is UP.
07/14/2005-06:53 Event Gateway (crypto operation) at port 11012 is UP.
07/14/2005-06:53 Event Gateway (crypto operation) at port 11014 is UP.
07/14/2005-06:53 Event Gateway (crypto operation) at port 11016 is UP.
07/14/2005-06:53 IMGW is UP.
07/14/2005-06:53 Cisco-CE Event Bus is UP.
07/14/2005-06:53 CEAdminService web service is DOWN in plain-text mode.
HTTP GET failed on URL http://infystorm5:80/cns/services/CEAdminService?wsdl
Connection refused

07/14/2005-06:53 CEConfigService web service is DOWN in plain-text mode.
HTTP GET failed on URL http://infystorm5:80/cns/services/CEConfigService?wsdl
Connection refused

07/14/2005-06:53 CEImageService web service is DOWN in plain-text mode.
HTTP GET failed on URL http://infystorm5:80/cns/services/CEImageService?wsdl
Connection refused

07/14/2005-06:53 CEExecService web service is DOWN in plain-text mode.
HTTP GET failed on URL http://infystorm5:80/cns/services/CEExecService?wsdl
Connection refused

07/14/2005-06:54 Exiting the CE-Health Beep Daemon.
```

Also, a configuration file (*/etc/logrotate.d/cnsce/ce_health*) is provided to rotate the above log file.

End User Interface

You can start, stop, and check the status of the daemon using the script called **MonitorCE**. This script is located in */etc/rc.d/init.d*. To know the status of Cisco Configuration Engine services, integrating applications have to issue the command:

/etc/rc.d/init.d/MonitorCE status

Usage

MonitorCE {start|stop|restart|reload|status}

- **start** – starts MonitorCE service. If MonitorCE service is already started, it does nothing.
- **stop** – stops MonitorCE service.
- **restart** – first stops the service, and then starts it again.
- **reload** – first stops the service, and then starts it again.
- **status** – tells if the service is up or not.

User Authentication

The Cisco Configuration Engine can authenticate a user by using the external authentication application. When a user logs into the Cisco Configuration Engine, instead of authenticating the user by using the Cisco Configuration Engine LDAP server, the Cisco Configuration Engine forwards the authentication request to an external authentication application. The Cisco Configuration Engine can support LDAP based authentication and integrate with the Microsoft Active Directory.

The Cisco Configuration Engine can authenticate the user both internally and externally based on the user selection during the Cisco Configuration Engine setup.

During the Cisco Configuration Engine setup, the administrator can select the authentication mode. The Cisco Configuration Engine prompts for IP address and user credentials for the remote LDAP server.

Choose the authentication mode of the system: *0=internal mode, 1=external mode*.

This example shows how to set the external authentication settings.

```
Enter IP Address of external directory server: 10.1.2.3
Enter port number of external directory server: [389]
Enter prefix for user name in external directory server: [cn]
Enter suffix for user name in external directory server: o=myorg,c=us
```

Also, the user can enable or disable the authorization.

This example shows how to set the external authorization settings:

```
Do you want to enable authorization? (y/n) [n] y
Enter UserDN for external directory server: cn=simpleuser,o=myorg,c=us
Enter password for the above user: *****
Re-enter password for the above user: *****
Enter role attribute name in user objectclass which defines the role: description
Enter role attribute value which defines the role of an administrator: administrator
```

Authorization

The Cisco Configuration Engine does not support task or resource-based authorization. However, the Cisco Configuration Engine GUI have Admin and Operator user levels. Depending on the role of the user, the appropriate GUI screens are displayed to the user. For more information about level of access, see Chapter 2, “Levels of Access”.

Backup Authentication-Authorization

To support existing the Cisco Configuration Engine users, backup authentication and authorization is supported for the external authentication mechanism. The Cisco Configuration Engine user who logs in is authenticated against the external application. If the external authentication fails, the user is authenticated against the Cisco Configuration Engine LDAP Server. The fall-back server will be the LDAP directory used by Cisco Configuration Engine (internal or external). If the user chooses internal authentication, Cisco Configuration Engine LDAP is used for authentication and there will be no fall-back authentication server used.

Multizone System Setup

The installation of the Cisco Configuration Engine software does not offer the multizone system setup by default. If you require a multizone system setup, you must enable the multizone feature during the system setup. To setup multiple IP addresses on the Cisco Configuration Engine server, you must manually customize the network parameters of the server to have multiple IP addresses. Multiple IP addresses can be configured by using IP aliasing on the network interface card. For more information see Chapter 6, “Setting Up a Multizone System” in *Cisco Configuration Engine Installation and Configuration Guide*.

Graphical User Interface

The Cisco Configuration Engine GUI is partially compliant with the Accessibility Design Requirements. This chapter provides general information about the GUI.

Logging In

Step 1 Launch your web browser.

This user interface supports:

- Internet Explorer 6.0 and above

Step 2 Go to the Cisco Configuration Engine URL.

For example: **http://<***ip_address***>**

Note If encryption is set during Setup (see "Encryption" section on page 1-12), use **https://<***ip_address***>**.

The login window appears (see Figure 2-1).

Figure 2-1 Logging Into the Configuration Engine

Step 3 Enter your **User ID**.

This is the value for the Configuration Engine login parameter that you entered during setup.

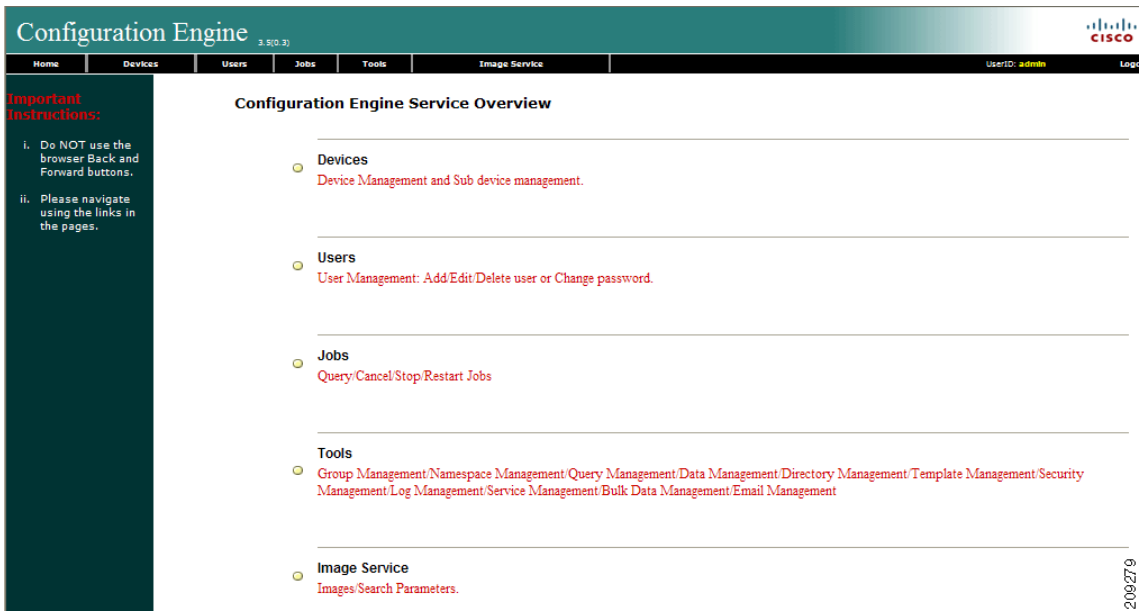
Step 4 Enter your password.

Step 5 Click **LOGIN**.

For an Administrator, the full-function Cisco Configuration Engine Home page appears (see Figure 2-2).

For an Operator, a limited-function Cisco Configuration Engine Home page appears without access to user-related tasks.

Figure 2-2 Administrator-level Home Page



Logging Out

To log out of the system, click **Logout**.

Levels of Access

In Internal Directory mode, there are two categories of users who have access to device information:

- Administrator
- Operator

An Administrator has full access to system administration tasks. An Operator has access to only limited set of tasks (see "Operator-Level Operations" section on page 2-25).

Operator-Level Operations

After logging into the Cisco Configuration Engine, an Operator has access to the following functions:

- Device
  - Add
  - Edit
  - Subdevices
  - Update Device
  - Query Device Inventory
- Tools
  - Change Password
  - View Event Log
  - View Image Server Log
- Jobs
  - Query Job
  - Cancel/Stop Job
  - Restart Job
- Image Service
  - View Image

Administrator-Level Operations

An Administrator can access all of the functions provided by the Cisco Configuration Engine user interface in both Internal Directory mode and External Directory mode.

Feature Operations

The Cisco Configuration Engine GUI (see Figure 2-2) provides the following feature operations:

- Devices – Click this tab to conduct operations on Devices and Subdevices (see Chapter 3, "Device and Subdevice Manager").
- Users – Click this tab to operate on user accounts (see Chapter 4, "User Account Manager").
- Jobs – Click this tab to access background update tasks that have been assigned a Job IDs (see Chapter 5, "Configuration and Image Update Jobs Manager").
- Tools – Click this tab to access the following features:
  - Group Manager (see Chapter 6, "Groups")
  - Namespace Manager (see Chapter 7, "Namespace Manager")
  - Query Manager (seeChapter 8, "Query Manager")
  - Data Manager (see Chapter 9, "Data Manager")
  - Directory Manager (see Chapter 10, "Directory Manager")
  - Parameter Manager (see Chapter 11, "Parameter Manager")
  - Template Manager (see Chapter 12, "Templates")
  - Security Manager (see Chapter 13, "Security Manager")
  - Log Manager (see Chapter 14, "Log Manager")
  - Service Manager (see Chapter 15, "Service Manager")
  - Bulk Data Manager (see Chapter 16, "Bulk Data Manager")
  - Email Manager (see Chapter 17, "Email Manager")
- Image Service – Click this tab to work with Images and Search Parameters (see Chapter 18, "Image Service").

Device and Subdevice Manager

To access Device tasks, log into the system (see "Logging In" section on page 2-23). Then, from the Home page, click the **Devices** tab.

The Device Functional Overview page appears showing:

- View Device
- Add Device
- Discover Device
- Edit Device
- Resynchronize Device
- Clone Device
- Delete Device
- Update Device
- Subdevices
- Query Device Inventory
- Delete Files on Device
- Dynamic Operations

Viewing Device Configuration

Step 1 From the Devices Functional Overview page, click **View Device**.

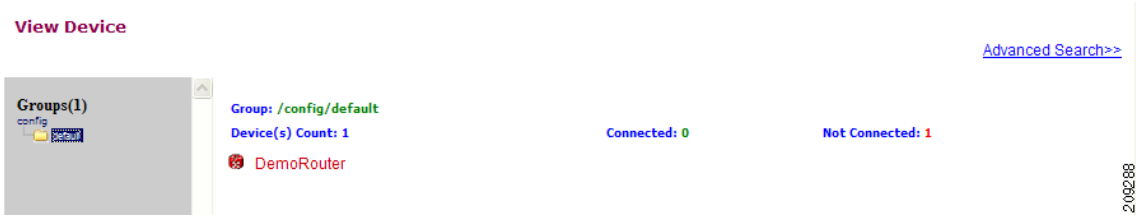
The Groups list appears.

Step 2 From the Groups list, select the group that holds the device you want to view.

Note You can also use the Advance Search feature on many GUI pages to locate devices based on user-define search parameters (see "Using Advanced Search Feature" section on page 3-30).

Step 3 The View Device list page appears (see Figure 3-1).

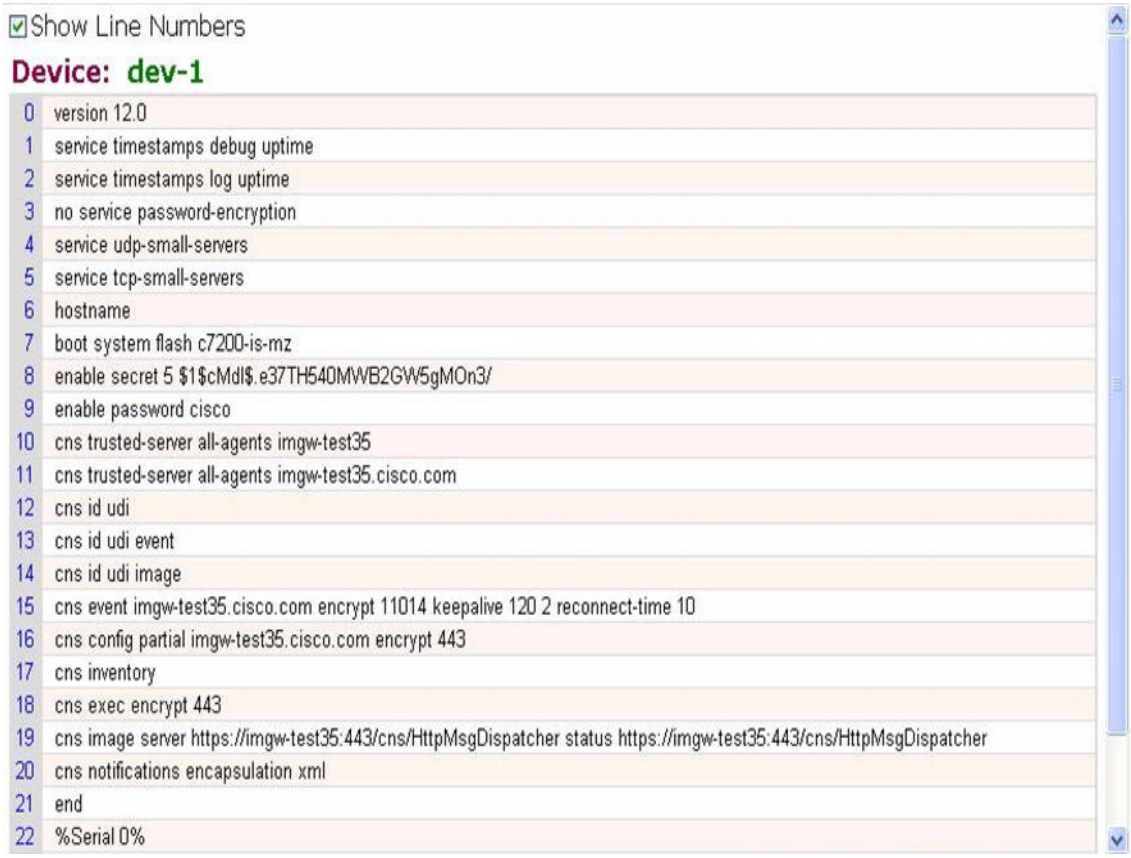
Figure 3-1 View Device List



Step 4 Click on the icon for the device you want to view.

The Configuration for that device appears (see Figure 3-2).

Figure 3-2 Device Configuration



Note The device configuration displayed is the configuration as it appears at the configuration server. It might not be the configuration running on the device.

Step 5 To view subdevices (if applicable), in the left navigation pane, click **View Subdevices.**

Step 6 To view Images associated with this device (if applicable), in the left navigation pane, click **View Images**.

Previewing Device Configuration

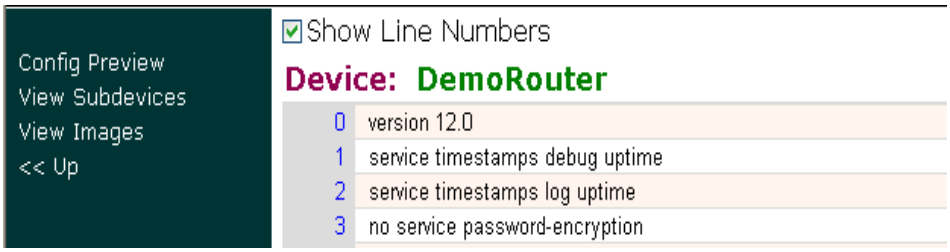
Step 1 From the Devices Functional Overview page, click **Edit Device**. The Groups list appears.

Step 2 From the Groups list, select the group that holds the device in question. The Edit Device list appears.

Step 3 Form the Edit Device list, select the group that holds the device you want to **Preview Device Configuration** or

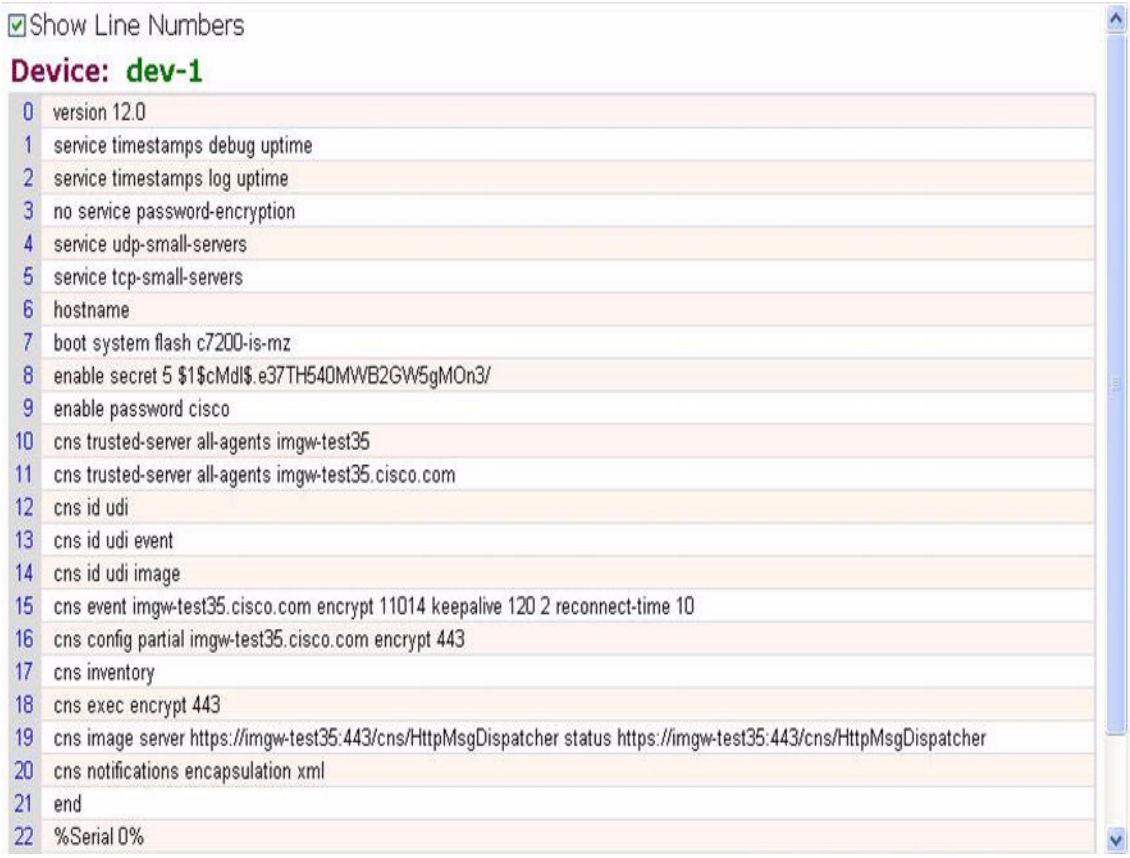
Step 4 From the Devices Functional Overview page, click **View Device**. The Groups list appears (see Figure 3-3).

Figure 3-3 Preview Device Configuration



Step 5 From the Groups list, select the group that holds the device you want to **Preview Device Configuration** (see Figure 3-4).

Figure 3-4 Device Configuration



Step 6 To preview subdevices configuration (if applicable), in the left navigation pane, click **View Subdevices**.

Using Advanced Search Feature

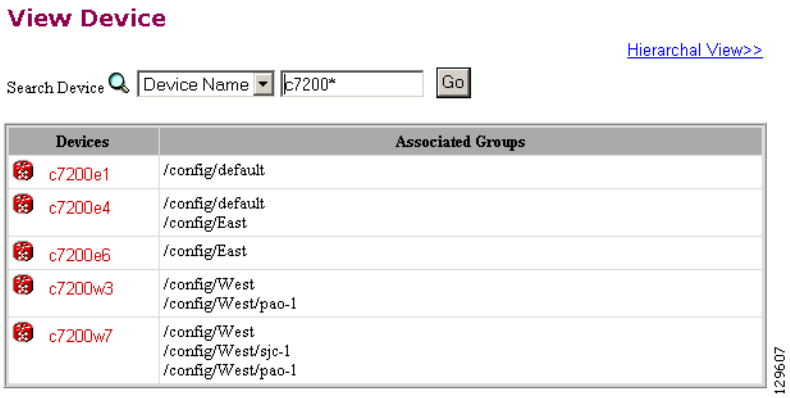
Step 1 From the Hierarchal View of groups (for example, see Figure 3-1), click **Advanced Search**.

Step 2 Use the drop-down arrow to select: **Config ID**, **Event ID**, or **Device Name** for the desired device.

Step 3 Then enter a value that corresponds to the first part of the argument, then click **Go**.

The results of the search are listed (see Figure 3-5).

Figure 3-5 Advanced Search Page



Adding Devices

There are three variations to the Add Device procedures based on **Device Type**:

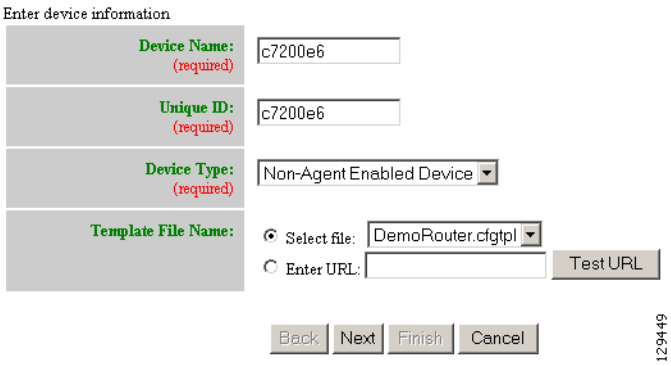
- Non-Agent Enabled Device (see below).
- Agent Enabled Device (see "Adding Agent Enabled Devices" section on page 3-39).
- PIX Firewall Device (see "Adding PIX Firewall Devices" section on page 3-44).
- ASA Firewall Device (see "Adding ASA Firewall Devices" section on page 3-47).

Adding Non-agent Enabled Devices

Step 1 From the Devices Functional Overview page, click **Add Device**.

The Device Information page appears (see Figure 3-6).

Figure 3-6 Device Information Page



Step 2 Enter a valid value (no spaces) in the **Device Name** field.

Table 3-1 shows valid values for these attributes.

Table 3-1 Valid Values for Add Device

Attribute	Description	Valid Values
Device Name	The name used as **cn** (common name) of the device.	a-z A-Z 0-9 -(hyphen) _ (under-score) . (period) :(colon)
Unique ID	Unique ID of the device.	Default or a-z A-Z 0-9 -(hyphen) _ (under-score) . (period) ,(comma) :(colon) /(forward-slash) =(equal) +(plus)
Device Type	Type of device	From drop-down list
Template File Name	Name of the configuration template to associate with the device.	From drop-down list, or user-defined

Step 3 In the **Unique ID** field, accept the default value that appears or enter another valid value (no spaces).

Step 4 For Device Type, from the drop-down list, select **Non-Agent Enabled Device**.

Step 5 Select the Template file name, then click **Next**.

The Group Membership page appears (see Figure 3-7).

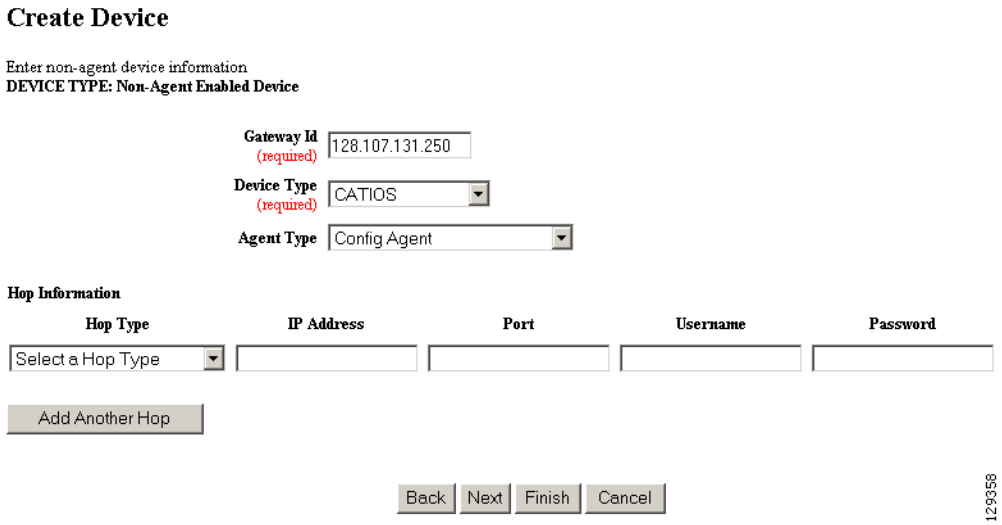
Figure 3-7 Group Membership

Tip Use the Group Manager to set up groups before you add a device (see "Creating Groups" section on page 6-96).

Step 6 Check to select the group(s) of which you want this device to become a member, then click **Next**.

The non-agent information (IMGW) page appears (see Figure 3-8).

Figure 3-8 Non-agent (IMGW) Information Page



Step 7 Enter the name of the device in the **Device Name** field.

Table 3-2 lists valid values for these fields.

Table 3-2 Valid Values for Add IMGW Device

Attribute	Description	Valid Values
Device Name	The name used as **cn** (common name) of the IMGW device.	Non-empty string excluding the special characters: !, “, #, $, %, &, ', (,), *, /, <, >, ?, @, \, ^, `, ~
Gateway ID	Gateway identifier for this device. This value is established during **Setup**. See *Cisco Configuration Engine Installation and Configuration Guide*.	Non-empty string excluding the special characters: !, “, #, $, %, &, ', (,), *, /, <, >, ?, @, \, ^, `, ~
Device Type	Type of IMGW device.	From drop-down list
Agent Type	Type of agent you want IMGW to simulate.	From drop-down list

Step 8 Enter the gateway ID in the **Gateway Id** field.

Note This value is established during **Setup**. See *Cisco Configuration Engine Installation and Configuration Guide*.

Step 9 Enter the appropriate Device and Hop information.

Tip Before you enter Hop information, see "Hop Tables" section on page 3-36.

Table 3-3 shows valid values for these fields.

Table 3-3 Valid Values for IMGW Device Hop Information

Attribute	Description	Valid Values
Hop Type	Type of IMGW hop.	From drop-down list
IP Address	IP address of the connecting node in the hop	Valid IP address of the following format: 10.1.14.216
Port	Port number of the node.	Integer values
Username	Username to login to the hop node.	String excluding the special characters: !, “, #, $, %, &, ', (,), *, /, <, >, ?, @, \, ^, `, ~
Password	Password to login to the hop node.	Non-null string

Step 10 To add another hop, click **Add Another Hop**, then enter hop information.

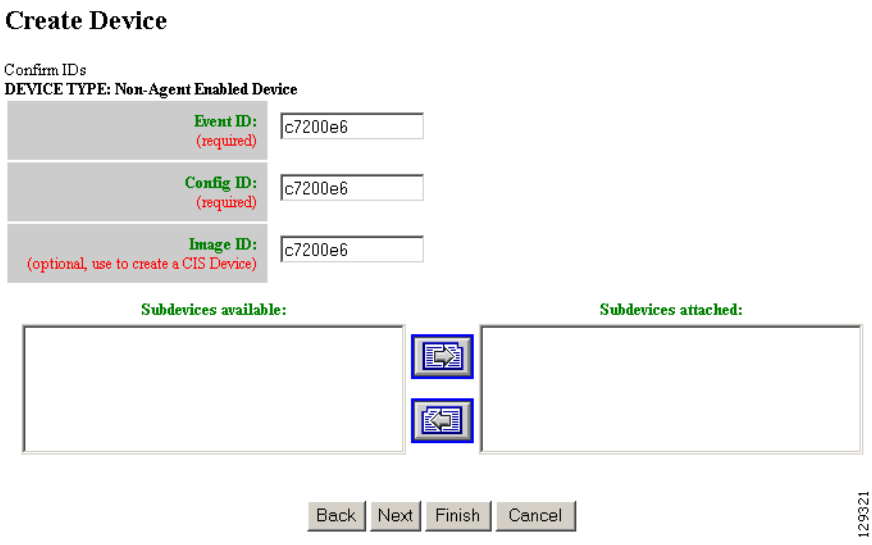
Step 11 To go back one page, click **Back**.

Step 12 To end this task, click **Finish**.

Step 13 To continue, click **Next**.

The Confirm IDs page appears

Figure 3-9 Confirm IDs Page



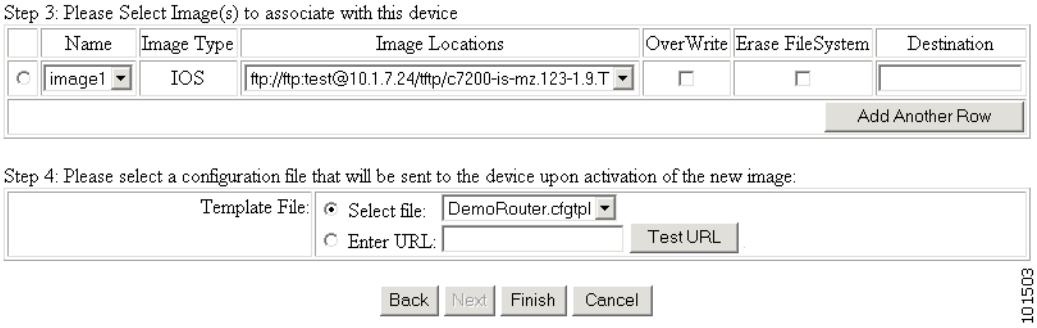
Step 14 To go back one page, click **Back**.

Step 15 To end this task, click **Finish**.

Step 16 To continue, click **Next**.

If you click **Next**, the Image Association page appears (see Figure 3-10).

Figure 3-10 Create Device > Image Association



Step 17 Select the image from the **Name** drop-down list.

The **Image Type** field and **Image Location** drop-down box are populated with corresponding information for the image.

Step 18 From the **Image Location** drop-down list, select the desired location.

Step 19 To add another row for image location, click **Add Another Row**.

You can locate multiple copies of an image on separate servers. This allows you to do load-sharing when updating a large number of devices. Each device in a large group can be associated with a copy of the image located at one of many server locations.

Step 20 In the Destination field, enter a valid URL where the image will be copied.

For example:

disk0:/c7200-mz

Step 21 To indicate which image is to be activated on the device after distribution, select the radio button in front of each row.

Step 22 Select the Configuration Control template file you want to send to this device for activation of a new image:

Tip Use the Configuration Control template that contains the CLI commands required for image activation for this device (see "Configuration Control Templates" section on page 12-127). If you do not have such a template, see "Adding a Template" section on page 12-138.

a. To select a template file from the drop-down list, click the **Select file** radio button.

b. Use the drop-down list to choose a template file.

OR

To use an external template:

a. Choose **Enter URL**.

b. Enter the full URL for the server, directory, and filename where the template is stored. Currently, only **http** is supported.

c. To test access to the external template, click **Test URL**.

If the server is unavailable or the external template cannot be accessed, an error appears. You can still save this logical device, but the template is not available until you have access to the external template.

Step 23 To clear this task, click **Cancel**.

Step 24 To go back to the previous page, click **Back**.

Step 25 To finish creating this device, click **Finish**.

Hop Tables

To access devices by means of Telnet, it is necessary to construct hop tables (see "HopInfo Examples" section on page 3-38). These are tables that indicate what network path exists to the device, and all the authentication information necessary at each stage, or hop.

What You Should Know About Device Hop Information

The Hop Information (HopInfo) structure describes one portion of the path between source and destination. HopInfo can be chained together to specify how to login to a device. Examples of uses of this structure include:

- Devices with basic authentication mode requiring IP address, username, and password
- Devices with additional authentication modes such as Cisco IOS enable mode
- Embedded-within-embedded applications such as line cards on a Catalyst switch

The latter two examples require a login, but not a hop to a different device. Therefore, they are referred to as *virtual* hops.

Table 3-4 shows the fields in the HopInfo structure:

Table 3-4 HopInfo Structure

Field	Purpose
hop_type	String indicating type of hop.
ip_address	IP address of device (string)
port	TCP port on which to access device (integer)
username	Username with which to login to device (string)
password	Password with which to login to device (string)

Currently Supported Device Types

Table 3-5 through Table 3-12 on page 3-38 provide the HopInfo list for devices that are directly accessible on the network by IMGW. For accessing devices by way of Commserver, see Table 3-13 on page 3-38.

All the rows in these tables are mandatory. Also, the hop_type fields cannot be NULL or empty. The fields marked with **X** are mandatory in IMGW unless they are not required on the device-side.

Table 3-5 Cisco IOS Device Directly Connected

hop_type	ip_address	port	username	password
IOS_LOGIN	X		X	X
IOS_EN			X	X

Table 3-6 Cisco IOS Device Directly Connected Supporting SSH

hop_type	ip_address	port	username	password
IOS_LOGIN:SSH	X		X	X
IOS_EN			X	X

Table 3-7 Catalyst Device Directly Connected

hop_type	ip_address	port	username	password
CATALYST_LOGIN	X		X	X
CATALYST_EN			X	X

Table 3-8 Catalyst IOS MSFC Blade Directly Connected

hop_type	ip_address	port	username	password
CATALYST_LOGIN	X		X	X
IOS_CAT_BLADE		X	X	X
IOS_EN			X	X

Table 3-9 Catalyst IOS Device Directly Connected

hop_type	ip_address	port	username	password
CATIOS_LOGIN	X		X	X
CATIOS_EN			X	X

Table 3-10 CSS Device Directly Connected

hop_type	ip_address	port	username	password
CSS_LOGIN	X		X	X
CSS_EN			X	X

Table 3-11 CE Device Directly Connected

hop_type	ip_address	port	username	password
CE_LOGIN	X		X	X
CE_EN			X	X

Table 3-12 PIX Device Directly Connected

hop_type	ip_address	port	username	password
PIX_LOGIN	X		X	X
PIX_EN			X	X

When any of the above devices is accessed by way of a Commserver (such as a Cisco 2511 Access Server), the resultant HopInfo list has the following two rows prepended to the respective HopInfo list for that device:

Table 3-13 Partial HopInfo List For Commserver Access

hop_type	ip_address	port	username	password
COMMSERVER_LOGIN	X		X	X
COMMSERVER		X	//////////////	X

Note Because the current release does not support port username, the username field of HopInfo structure for COMMSERVER is always ignored by IMGW. Do not set up the port username on the Commserver.

HopInfo Examples

Table 3-14 Cisco IOS Device Directly Connected

hop_type	ip_address	port	username	password
IOS_LOGIN	172.28.6.90		Johndoe	Passnow
IOS_EN			dummy	compass

Table 3-15 Cisco IOS Device Directly Connected Supporting SSH

hop_type	ip_address	port	username	password
IOS_LOGIN:SSH	172.28.6.90		Johndoe	Passnow
IOS_EN			dummy	compass

Table 3-16 Cisco IOS Device Connected With Commserver

hop_type	ip_address	port	username	password
COMMSERVER_LOGIN	172.28.6.226		Sandra	Me1100
COMMSERVER		2005	//////////////////	Lab123
IOS_LOGIN			Johndoe	Passnow
IOS_EN			dummy	compass

Table 3-17 Catalyst IOS MFSC Blade Directly Connected

hop_type	ip_address	port	username	password
CATALYST_LOGIN	172.29.132.32		Admin	Raining
IOS_CAT_BLADE		15	Admin	winding
IOS_EN			dummy	moonlight

Table 3-18 Catalyst IOS MFSC Blade Accessed With Commserver

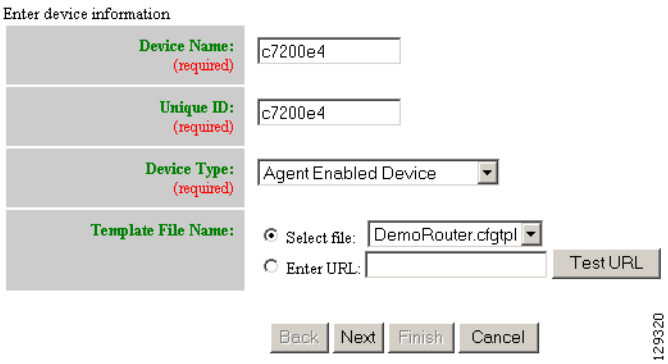
hop_type	ip_address	port	username	password
COMMSERVER_LOGIN	172.28.22.229		Kldfg	Dsdsfg
COMMSERVER		2010	//////////////////	Dadada
CATALYST_LOGIN			Admin	Raining
IOS_CAT_BLADE		15	Admin	winding
IOS_EN			dummy	moonlight

Adding Agent Enabled Devices

Step 1 From the Devices Functional Overview page, click **Add Device**.

The Device Information page appears (see Figure 3-11).

Figure 3-11 Device Information Page



Step 2 Enter a valid value (no spaces) in the **Device Name** field.

Table 3-19 shows valid values for these attributes.

Table 3-19 Valid Values for Add Device

Attribute	Description	Valid Values
Device Name	The name used as **cn** (common name) of the device.	a-z A-Z 0-9 -(hyphen) _ (under-score) . (period)
Unique ID	Unique ID of the device.	Default or a-z A-Z 0-9 -(hyphen) _ (under-score) . (period)
Device Type	Type of device	From drop-down list
Template File Name	Name of the configuration template to associate with the device.	From drop-down list, or user-defined

Step 3 In the **Unique ID** field, accept the default value that appears or enter another valid value (no spaces).

Step 4 For Device Type, from the drop-down list, select **Agent Enabled Device**.

Step 5 Select the Template file name, then click **Next**.

Note To associate an external template to this device, select **Enter URL** with the appropriate path.

The Group Membership page appears (see Figure 3-12).

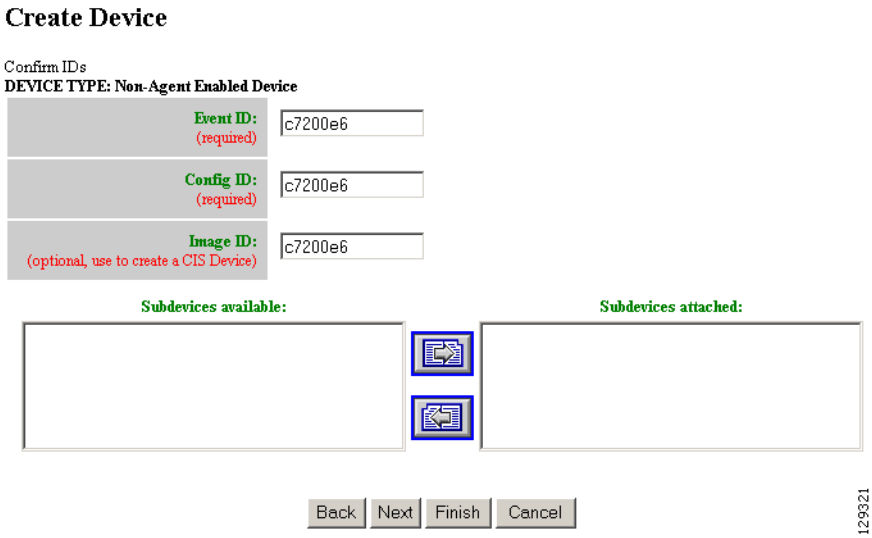
Figure 3-12 Group Membership Page

Tip Use the Group Manager to set up groups before you add a device (see "Creating Groups" section on page 6-96).

Step 6 Check to select the group(s) of which you want this device to become a member, then click **Next**.

The device IDs page appears (see Figure 3-13).

Figure 3-13 Device IDs Page



Step 7 Enter the appropriate IDs.

Table 3-20 shows valid values for these attributes.

Table 3-20 Valid Values for Agent Enabled Device IDs

Attribute	Description	Valid Values
Event ID	Event ID to be associated with this device.	Default, or a-z A-Z 0-9 -(hyphen) _ (under-score) . (period) ,(comma) :(colon) /(forward-slash) =(equal) +(plus)
Config ID	Configuration ID to be associated with this device.	Default, or a-z A-Z 0-9 -(hyphen) _ (under-score) . (period), (comma) :(colon) /(forward-slash) =(equal) +(plus)
Image ID	Image ID to be associated with this device.	Default, or a-z A-Z 0-9 -(hyphen) _ (under-score) . (period) ,(comma) :(colon) /(forward-slash) =(equal) +(plus)

Step 8 If applicable, select and assign subdevices to this device.

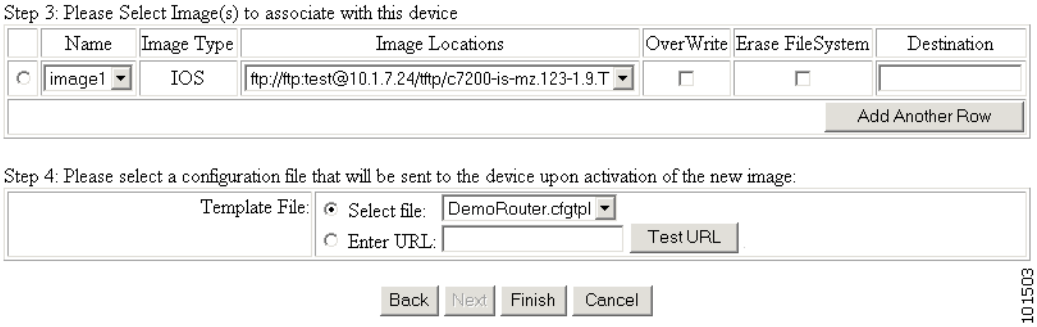
Step 9 To go back one page, click **Back**.

Step 10 To end this task, click **Finish**.

Step 11 To continue by associating this device with an image, click **Next**.

If you click **Next**, the Image Association page appears (see Figure 3-14).

Figure 3-14 Create Device > Image Association



Step 12 Select the image from the **Name** drop-down list.

The **Image Type** field and **Image Location** drop-down box are populated with corresponding information for the image.

Step 13 From the **Image Location** drop-down list, select the desired location.

Step 14 To add another row for image location, click **Add Another Row**.

You can locate multiple copies of an image on separate servers. This allows you to do load-sharing when updating a large number of devices. Each device in a large group can be associated with a copy of the image located at one of many server locations.

Step 15 In the Destination field, enter a valid URL where the image will be copied.

For example:

disk0:/c7200-mz

Step 16 To indicate which image is to be activated on the device after distribution, select the radio button in front of each row.

Step 17 Select the Configuration Control template file you want to send to this device for activation of a new image:

Tip Use the Configuration Control template that contains the CLI commands required for image activation for this device (see "Configuration Control Templates" section on page 12-127). If you do not have such a template, see "Adding a Template" section on page 12-138.

a. To select a template file from the drop-down list, click the **Select file** radio button.

b. Use the drop-down list to choose a template file.

OR

To use an external template:

a. Choose **Enter URL**.

b. Enter the full URL for the server, directory, and filename where the template is stored. Currently, only **http** is supported.

c. To test access to the external template, click **Test URL**.

If the server is unavailable or the external template cannot be accessed, an error appears. You can still save this logical device, but the template is not available until you have access to the external template.

Step 18 To clear this task, click **Cancel**.

Step 19 To go back to the previous page, click **Back**.

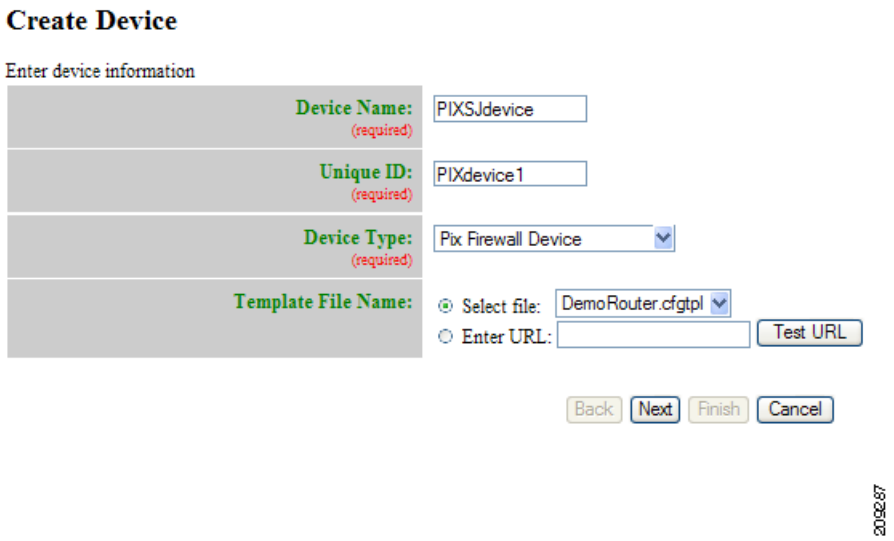
Step 20 To finish creating this device, click **Finish**.

Adding PIX Firewall Devices

Step 1 From the Devices Functional Overview page, click **Add Device**.

The Device Information page appears (see Figure 3-15).

Figure 3-15 Device Information Page



Step 2 Enter a valid value (no spaces) in the **Device Name** field.

Table 3-21 shows valid values for these attributes.

Table 3-21 Valid Values for Add Device

Attribute	Description	Valid Values
Device Name	The name used as **cn** (common name) of the device.	a-z A-Z 0-9 -(hyphen) _ (under-score) . (period)
Unique ID	Unique ID which is configured on the device.	Default or a-z A-Z 0-9 -(hyphen) _ (under-score) . (period)
Device Type	Type of device	From drop-down list
Template File Name	Name of the configuration template to associate with the device.	From drop-down list, or user-defined

Step 3 In the **Unique ID** field, accept the default value that appears or enter another valid value (no spaces).

Step 4 For Device Type, from the drop-down list, select **PIX Firewall Device**.

Step 5 Select the Template file name, then click **Next**.

The Group Membership page appears (see Figure 3-16).

Figure 3-16 Group Membership Page

Tip Use the Group Manager to set up groups before you add a device (see "Creating Groups" section on page 6-96).

Step 6 Check to select the group(s) of which you want this device to become a member, then click **Next**.

The PixAuthentication Password page appears (see Figure 3-17).

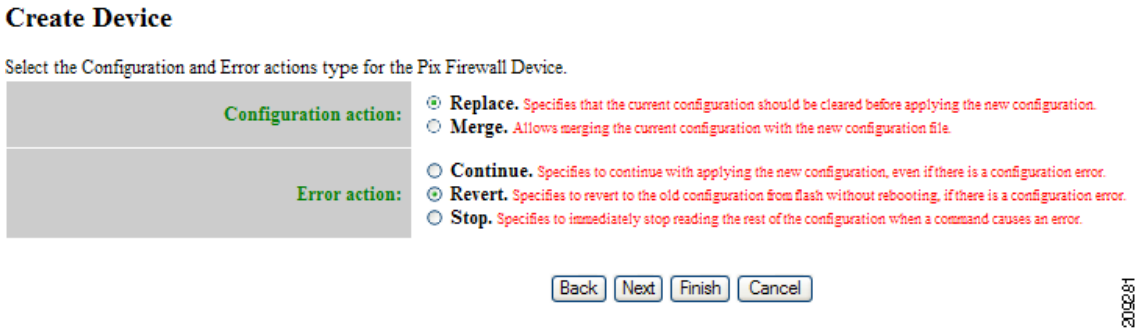
Figure 3-17 PIX Authentication Password Page

Step 7 Enter authentication password for PIX devices.

A case-sensitive password of up to 16 alphanumeric and special characters. Any character can be used in the password except a question mark and a space.

Step 8 Click the **Next** button. The PIX **Configuration and Error Actions Type** page appears (see Figure 3-18).

Figure 3-18 PIX Configuration and Error Actions Type Page



Step 9 From the **Configuration and Error Actions Type** page, choose the appropriate options (Replace, Merge, Continue, Revert, and Stop).

Step 10 To go back one page, click **Back**.

Step 11 To end this task, click **Finish**.

Step 12 To continue by associating this device with an image, click **Next**.

Step 13 If you click **Next**, the Image Association page for PIX Firewall Devices appears.

Step 14 Select the image from the **Name** drop-down list.

The **Image Type** field and **Image Location** drop-down box are populated with corresponding information for the image.

Note Only PIX or PDM images can be associated with a PIX device.

Step 15 From the **Image Location** drop-down list, select the desired location.

Step 16 To add another row for image location, click **Add Another Row**.

Note For PIX devices, you can have only one PIX image and one PDM image.

Step 17 To indicate whether the image is to be activated on the device after distribution, check the box in front of each row.

Step 18 To cancel creating a device and return to the Devices main menu, click **Cancel**.

Step 19 To go back to the previous page, click **Back**.

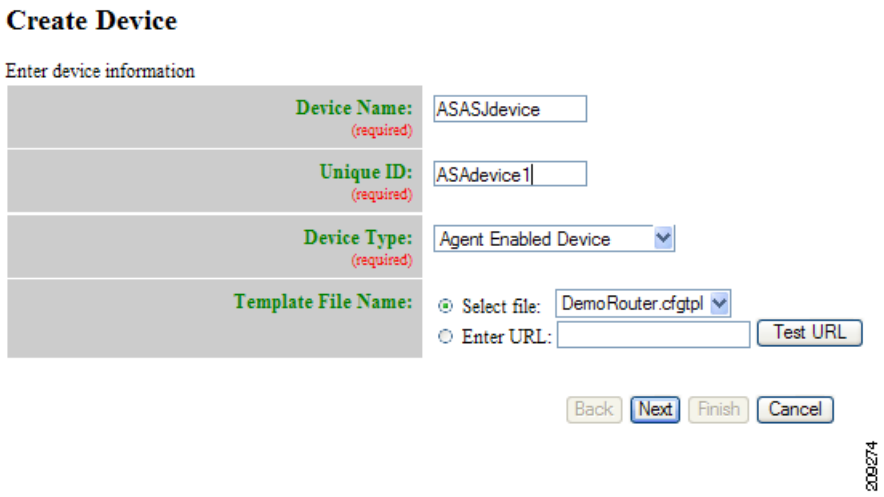
Step 20 To finish creating this device, click **Finish**.

Adding ASA Firewall Devices

Step 1 From the Devices Functional Overview page, click **Add Device**.

The Device Information page appears (see Figure 3-19).

Figure 3-19 Device Information Page



Step 2 Enter a valid value (no spaces) in the **Device Name** field.

Table 3-22 shows valid values for these attributes.

Table 3-22 Valid Values for Add Device

Attribute	Description	Valid Values
Device Name	The name used as **cn** (common name) of the device.	a-z A-Z 0-9 -(hyphen) _ (under-score) . (period)
Unique ID	Unique ID which is configured on the device.	Default or a-z A-Z 0-9 -(hyphen) _ (under-score) . (period)
Device Type	Type of device	From drop-down list
Template File Name	Name of the configuration template to associate with the device.	From drop-down list, or user-defined

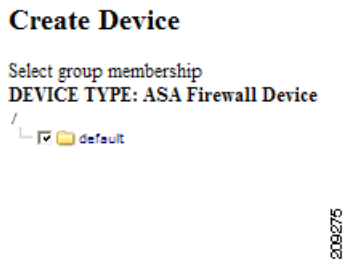
Step 3 In the **Unique ID** field, accept the default value that appears or enter another valid value (no spaces).

Step 4 For Device Type, from the drop-down list, select **ASA Firewall Device**.

Step 5 Select the Template file name, then click **Next**.

The Group Membership page appears (see Figure 3-20).

Figure 3-20 Group Membership Page

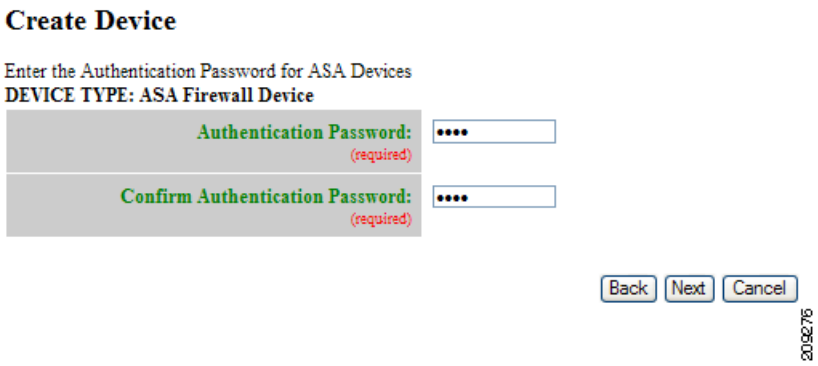


Tip Use the Group Manager to set up groups before you add a device (see "Creating Groups" section on page 6-96).

Step 6 Check to select the group(s) of which you want this device to become a member, then click **Next**.

The ASA Authentication Password page appears (see Figure 3-21).

Figure 3-21 ASA Authentication Password Page

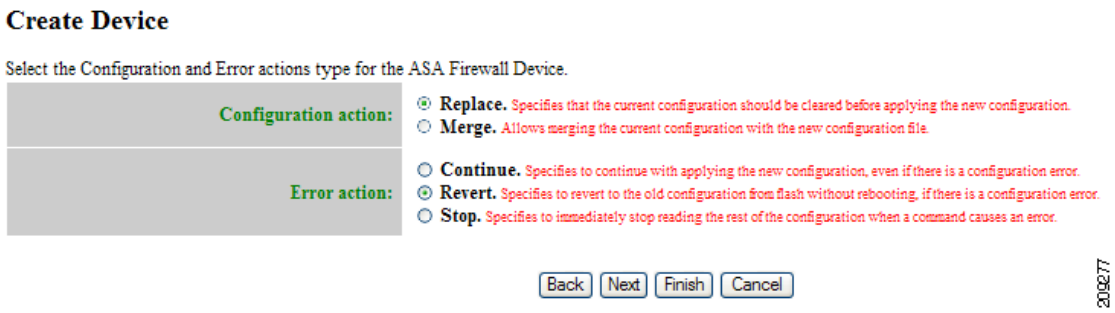


Step 7 Enter authentication password for ASA devices.

A case-sensitive password of up to 16 alphanumeric and special characters. Any character can be used in the password except a question mark and a space.

Step 8 Click the **Next** button. The ASA **Configuration and Error Actions Type** page appears (see Figure 3-17).

Figure 3-22 ASA Configuration and Error Actions Type Page



Step 9 From the **Configuration and Error Actions Type** page, choose the appropriate options (Replace, Merge, Continue, Revert, and Stop).

Step 10 To go back one page, click **Back**.

Step 11 To end this task, click **Finish**.

Step 12 To continue by associating this device with an image, click **Next**.

Step 13 If you click **Next**, the Image Association page for PIX Firewall Devices appears.

Step 14 Select the image from the **Name** drop-down list.

The **Image Type** field and **Image Location** drop-down box are populated with corresponding information for the image.

Note Only ASA or ASDM images can be associated with a ASA device.

Step 15 From the **Image Location** drop-down list, select the desired location.

Step 16 To add another row for image location, click **Add Another Row**.

Note For ASA devices, you can have only one ASA image and one ASDM image.

Step 17 To indicate whether the image is to be activated on the device after distribution, check the box in front of each row.

Step 18 To cancel creating a device and return to the Devices main menu, click **Cancel**.

Step 19 To go back to the previous page, click **Back**.

Step 20 To finish creating this device, click **Finish**.

Discovering Devices

Cisco Configuration Engine can discover a device once the device (for this example: **router-3460**) is configured for CNS. For more information about this, see *CNS Image Agent* at:

http://www.cisco.com/en/US/docs/net_mgmt/configuration_engine/3.5/installation/guide/CE_3_ig_security.html

During the execution of **setup.sh** for the Cisco Configuration Engine host, the settings configured would be:

```
...
For detail information about the parameters in this setup, refer to "Cisco Configuration
Engine Administration Guide."
...

Encryption settings:
--------------------
Enable cryptographic (crypto) operation between Event Gateway(s)/Config server and
device(s) (y/n)? n
Each Event Gateway process serves 500 devices. Maximum number of
Event Gateways allowed is 20.
Enter number of Event Gateways that will be started with crypto operation:[1] 0
Enter number of Event Gateways that will be started with plaintext operation: [5] 2
Enter Cisco-CE Event Bus Network Parameter: [ce_host_hostname or ce_host_ip_address]
```

Note For more information about running **setup.sh**, see the *Cisco Configuration Engine Installation and Configuration Guide*.

Step 1 Log in to **router-3460**

Step 2 Using the Cisco IOS CLI command: **show running configuration**, verify that **router-3460** is configured with IP routing. For example:

```
hostname router-3460
...
ip cef
ip host ce_host 10.1.2.3
...
interface Ethernet0/0
ip address 10.1.2.4 255.255.255.0
...
ip default-gateway 10.1.2.1
```

```
...
ip classless
ip route 0.0.0.0 0.0.0.0 10.1.2.1
```

where:

router-3460 is the hostname identifying the device for Cisco Configuration Engine and 10.1.2.3 is the IP address of the Cisco Configuration Engine.

Step 3 Log in to **router-3640** and perform the following operations:

configure terminal ip host ce_host 10.1.2.3
cns trusted-server all-agents ce_host
cns id string router-3460
cns id string router-3460 event
cns event ce_host 11013
cns config notify all interval 1 old-format
cns config partial ce_host 80
cns exec 80

Note The above configuration will support Discover Device as well as downloading a configuration, which requires **cns config partial ce_host 80**.

Step 4 Verify IP connectivity between **ce_host** and **router-3640** by issuing the **ping** command from **ce_host** to **router-3640** and from **router-3640** to **ce_host**.

Step 5 Create a template.

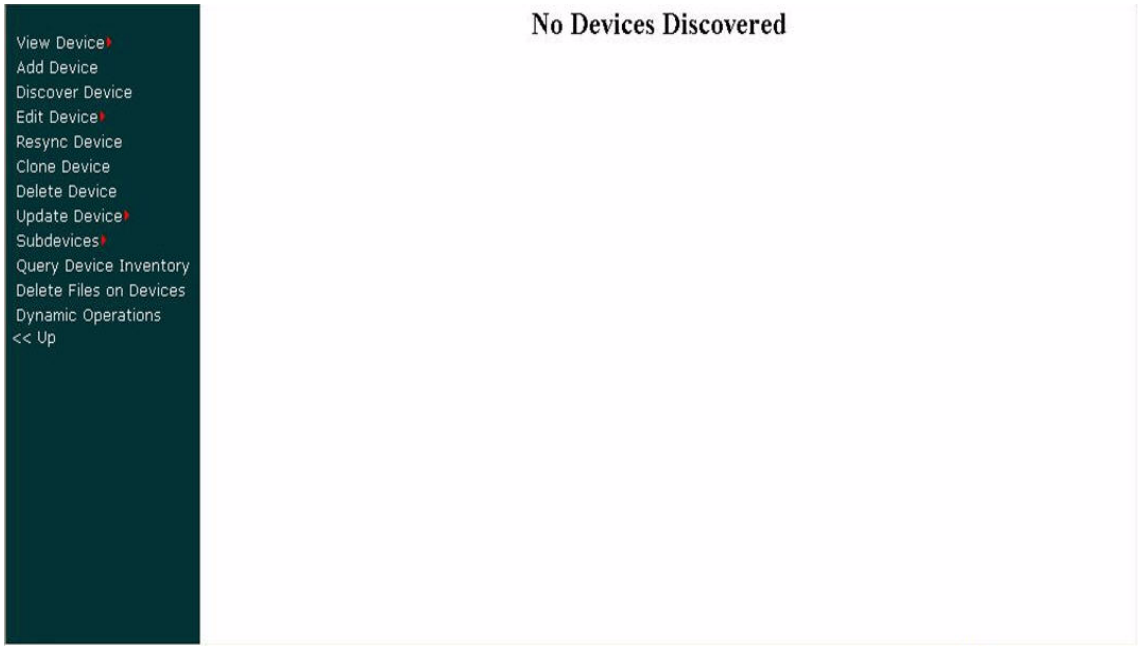
For our example, name it **router-3460**.

You must insert a minimum of one line in the template. You can add a **!** for this.

Note For more information about creating a template, see Chapter 12, "Templates."

Step 6 On the Device Functional Overview page, choose **Discover Device**.

Figure 3-23 Discover Device Page



When the discovery task completes, the following information appears:

```
Discover Devices
There are 1 device(s) currently connected to the IE2100 but not yet created in the
directory.
Select the devices you want to create and click on 'Create'.
Device Name DeviceID Connected Time Template Name Group Name
router-3640 router-3640 1/19/06 9:46:03 AM
- DemoRouter.cfgtpl
- Acquire Running Config /config/default
- Acquire Startup Config
```

Step 7 Click on the check box for **router-3640**, then click on the radio button and move the cursor to **router-3640.cfgtpl**.

Step 8 Choose **Create**.

The following information appears:

```
Status of Discovered Device Creation:
Device Name Template Name Status
router-3640 router-3640.cfgtpl Success
```

Step 9 On the Device Functional Overview page, choose **View Device**.

You should see an icon for **router-3640**.

The icon color should be green indicating communication between **ce_host** and **router-3640** has been established.

Notes:

1. Before a device is discovered or created, we recommend that you configure a template for the device. When Cisco Configuration Engine discovers a device, or you create a device, you then must associate the device with a template. Although Cisco Configuration Engine has a default sample template (DemoRouter.cfgtpl) already created, it is very unlikely that your device will be configured using DemoRouter.cfgtpl. Therefore, create a new template.
2. If **Create Device** is performed after configuring a template for **router-c3460**, then Cisco Configuration Engine will not discover this router (you will not see an icon for **router-c3460** when Discover Device is selected). If you want Cisco Configuration Engine to discover the device then create only a template for the device—DO NOT use the **Create Device** operation. If you use **Create Device**, and you go to **Discover Device**, you will not see an icon for **router-c3460**. However, in either case, **View Device** should show an icon for **router-c3460**.
3. The Cisco Configuration Engine host uses odd numbered event ports for messages sent in plain text. For example, the default Cisco Configuration Engine setting is **5** event gateway ports without crypto enabled. Devices use ports 11013, 11015, 11017, 11019, 11021 depending on what you configured on the device (for **cns event 10.1.2.3 11013** this means event gateway port 11013 is used by **router-c3640** to communicate with the Cisco Configuration Engine host, 10.1.2.3).
4. The Cisco Configuration Engine host uses even numbered event ports for message sent encrypted starting with 11014. For example, if you set the number of event gateways to **2** during setup, then ports 11014 and 11016 would be available for use by a device.

Note

- The ports for Event Gateways with crypto operation are even numbers that start from 11012.
- The ports for Event Gateways with plaintext operation are odd numbers that start from 11011.

Editing Devices

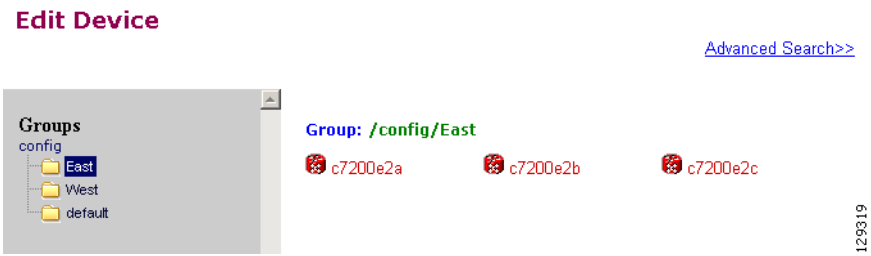
Step 1 From the Devices Functional Overview page, click **Edit Device**.

The Groups list appears.

Step 2 From the Groups list, select the group that holds the device in question.

The Edit Device list appears (see Figure 3-24).

Figure 3-24 Edit Device List



Step 3 Click on the icon for the device you want to edit. The device configuration appears (see Figure 3-25).

Figure 3-25 Device Configuration

Show Line Numbers

Device: dev-1

```
0  version 12.0
1  service timestamps debug uptime
2  service timestamps log uptime
3  no service password-encryption
4  service udp-small-servers
5  service tcp-small-servers
6  hostname
7  boot system flash c7200-is-mz
8  enable secret 5 $1$cMdI$.e37TH540MWB2GW5gMOn3/
9  enable password cisco
10 cns trusted-server all-agents imgw-test35
11 cns trusted-server all-agents imgw-test35.cisco.com
12 cns id udi
13 cns id udi event
14 cns id udi image
15 cns event imgw-test35.cisco.com encrypt 11014 keepalive 120 2 reconnect-time 10
16 cns config partial imgw-test35.cisco.com encrypt 443
17 cns inventory
18 cns exec encrypt 443
19 cns image server https://imgw-test35:443/cns/HttpMsgDispatcher status https://imgw-test35:443/cns/HttpMsgDispatcher
20 cns notifications encapsulation xml
21 end
22 %Serial 0%
```

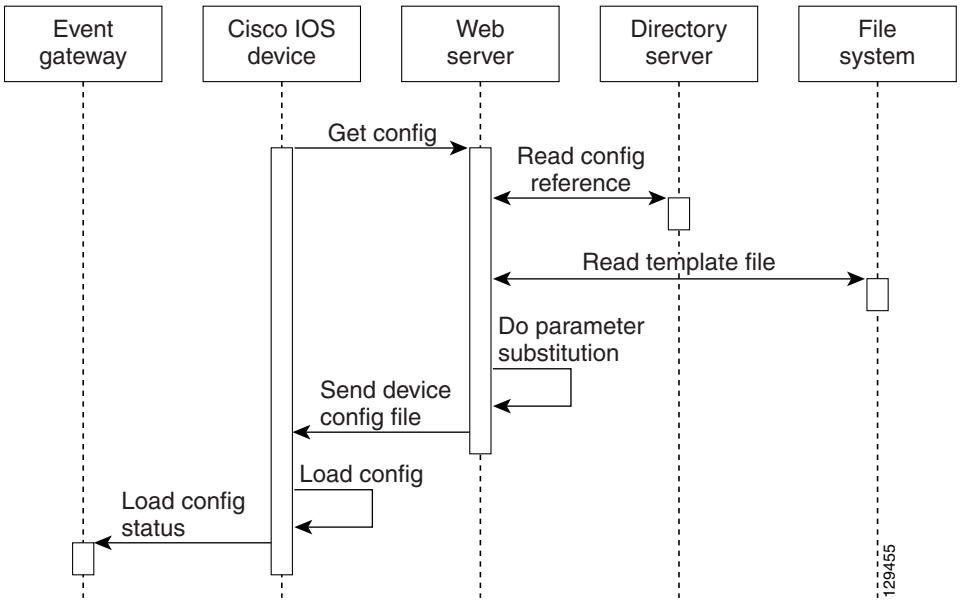
Step 4 From the left navigation pane, choose the edit function you want to use.

Editing Non-agent Enabled Device Information

Step 1 From the Edit Device page, click **Edit Information**.

The device information editor page appears (see Figure 3-26).

Figure 3-26 Non-agent Device Information Editor

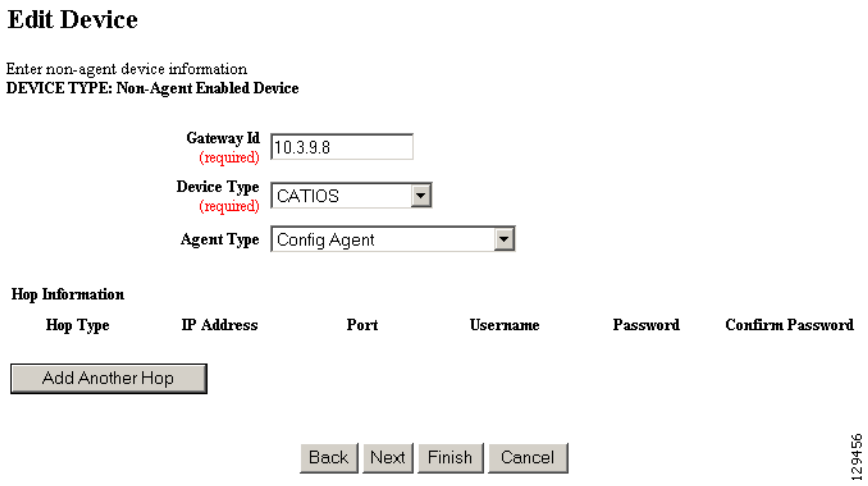


Step 2 To modify the device name, enter a valid value (no spaces) in the **Device Name** field, then click **Next**.

Step 3 Select Group Membership, then click **Next**.

The Non-agent Edit Device Information page appears (see Figure 3-27).

Figure 3-27 Non-agent Information Page

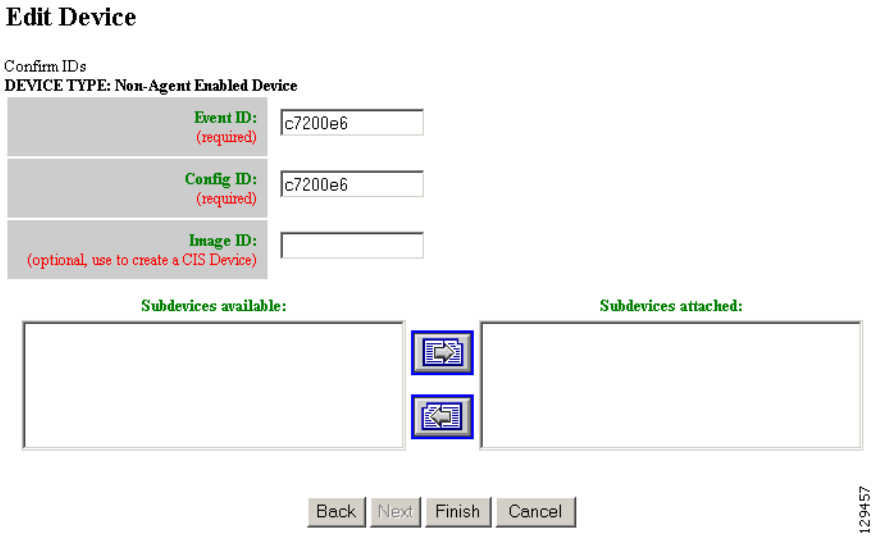


Step 4 Edit all appropriate fields, then to end this task, click **Finish**.

Step 5 To continue, click **Next**.

The device IDs page appears (see Figure 3-28).

Figure 3-28 Edit Non-agent Device IDs Page



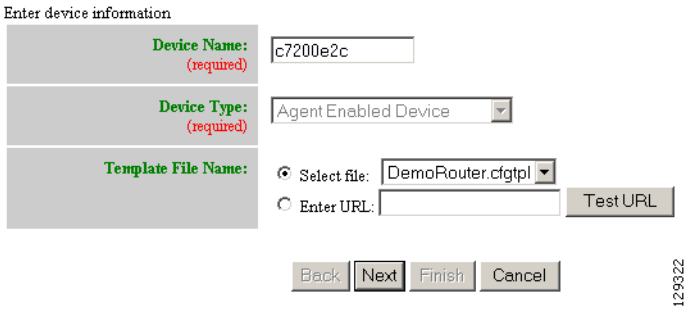
Step 6 Modify devices IDs as required, then click **Finish**.

Editing Agent Enabled Device Information

Step 1 From the Edit Device page, click **Edit Information**.

The device information editor page appears (see Figure 3-29).

Figure 3-29 Agent Enabled Device Information Page

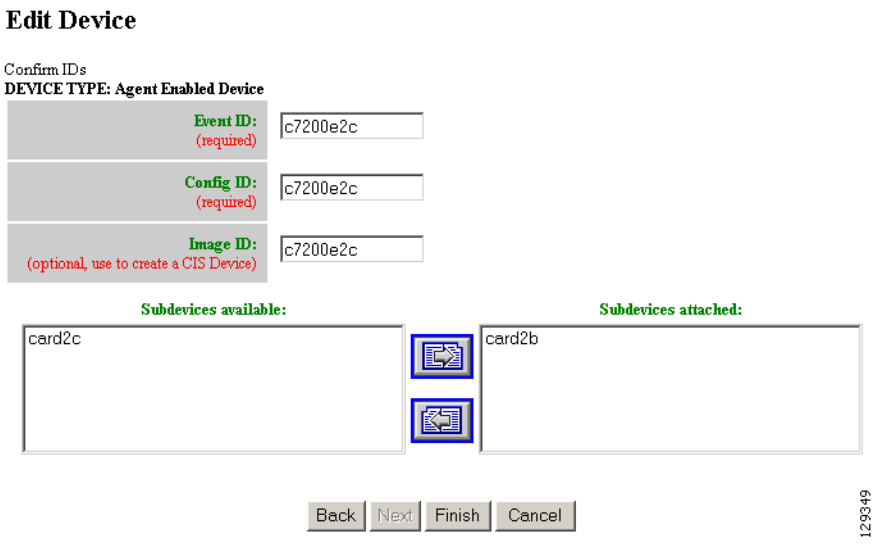


Step 2 To modify the device name, enter a valid value (no spaces) in the **Device Name** field, then click **Next**.

Step 3 Select Group Membership, then click **Next**.

The device IDs page appears (see Figure 3-30).

Figure 3-30 Agent enabled Device IDs Page



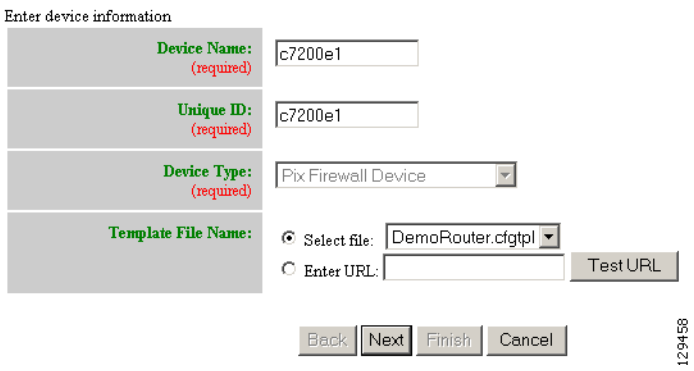
Step 4 Modify device IDs as required, then click **Finish**.

Editing PIX Device Information

Step 1 From the Edit Device page, click **Edit Information**.

The device information editor page appears (see Figure 3-31).

Figure 3-31 PIX Device Information Page

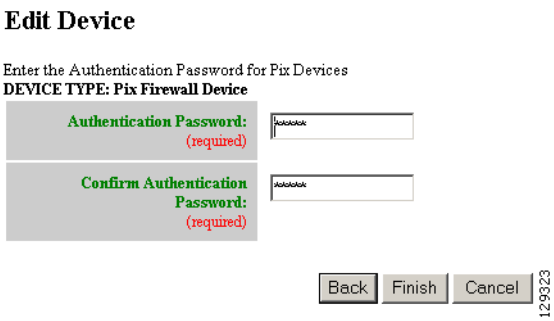


Step 2 To modify the device name and Image ID, if applicable, then click **Next**.

Step 3 Select Group Membership, then click **Next**.

The PIX Device Authentication Password page appears, see Figure 3-32.

Figure 3-32 PIX Device Authentication Password



Step 4 Modify the authentication password if required, then click **Finish**.

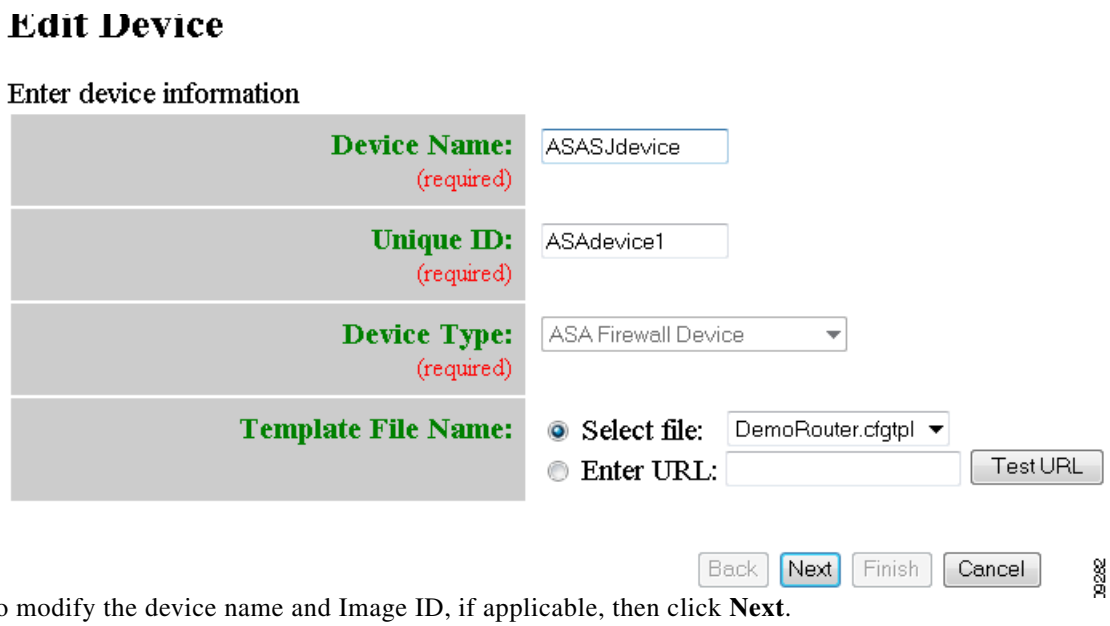
A case-sensitive password of up to 16 alphanumeric and special characters. Any character can be used in the password except a question mark and a space.

Editing ASA Device Information

Step 1 From the Edit Device page, click **Edit Information**.

The device information editor page appears (see Figure 3-33).

Figure 3-33 ASA Device Information Page

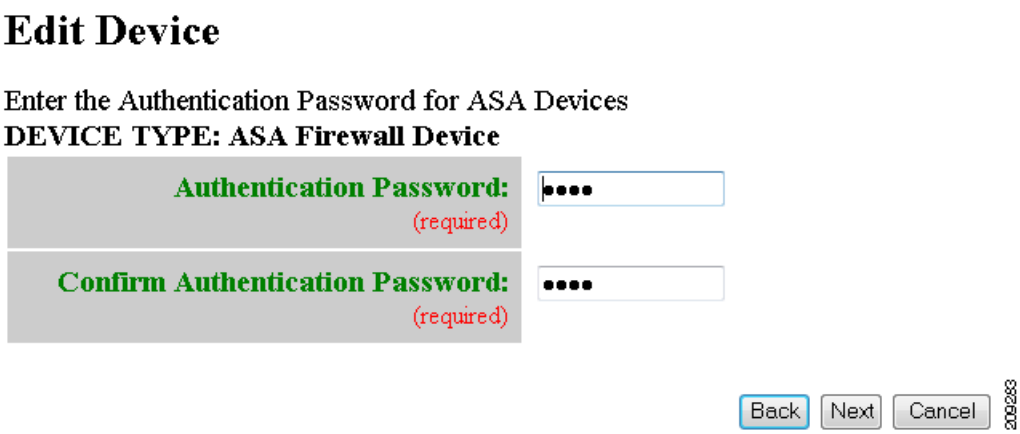


Step 2 To modify the device name and Image ID, if applicable, then click **Next**.

Step 3 Select Group Membership, then click **Next**.

The ASA Device Authentication Password page appears, see Figure 3-34.

Figure 3-34 ASA Device Authentication Password



Step 4 Modify the authentication password if required, then click **Finish**.

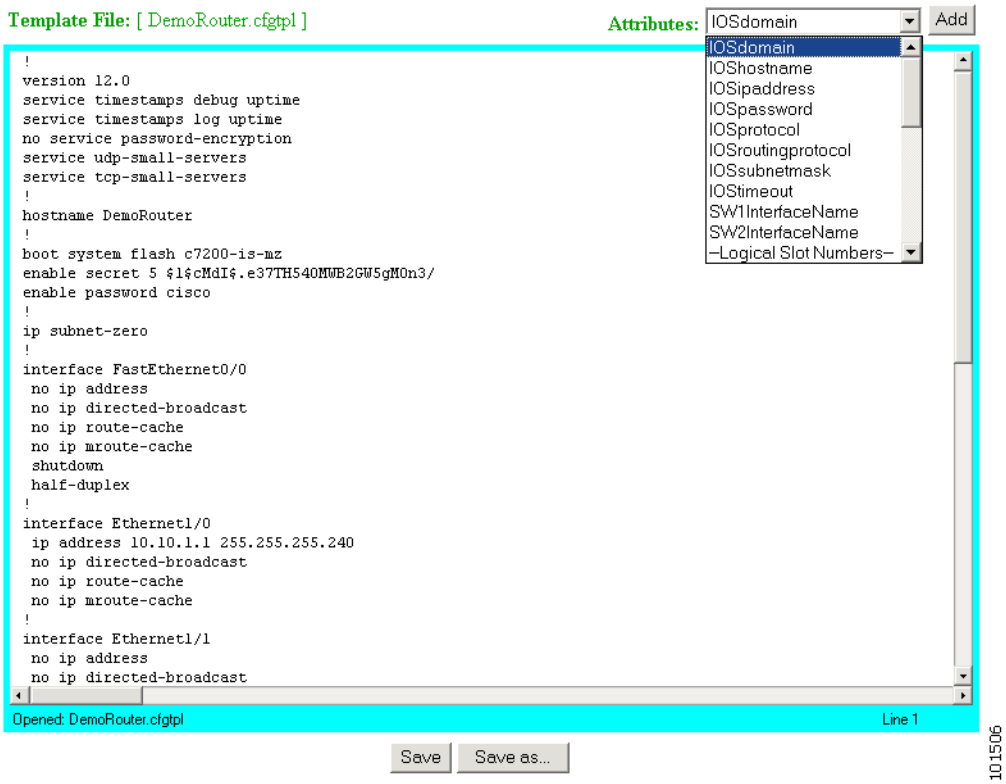
A case-sensitive password of up to 16 alphanumeric and special characters. Any character can be used in the password except a question mark and a space.

Editing Device Templates

Step 1 From the Edit Device page, click **Edit Template**.

The template editor appears (see Figure 3-35).

Figure 3-35 Template Editor



Step 2 In the **Attributes** field, click the drop-down arrow.

Step 3 Choose the attribute you want to add to the template, then click **Add**.

Step 4 Repeat Steps 2 and 3 for all attributes you want to add to the template file.

Step 5 Delete all unusable strings from the template file.

Step 6 Edit strings as necessary.

The default multi-line begin and end tags are **^C** and **^C** respectively. The delimiter for these tags are: ~ ! @ ^ & * - = |. Do not use # or %.

For example, a multi-line test banner might be:

```
banner exec ^C
    This is a Test Banner
    1. Hi
    2. Hello
    3. Test is 1234567890
^C
```

Step 7 To save your edits, click **Save**.

Step 8 To save this version as a new template, click **Save as**.

Editing Device Parameters

Step 1 From the Edit Device page:

a. If you have administrator-level access click **Edit Parameter-admin**.

b. To use Operator-level access click **Edit Parameter-operator**.

The parameters editor appears.

Step 2 Edit all active lines as required.

Step 3 To save your edits, click **Save Parameters**.

Editing Contact Information

Step 1 From the Edit Device page, click **Edit ContactInfo**.

The contact information appears.

Step 2 Edit all active fields as required.

Step 3 To clear your entries, click **Reset**.

Step 4 To save your edits, click **Update**.

Editing Subdevices

For complete information about working with subdevices, including editing (except PIX devices), see "Working with Subdevices" section on page 3-71.

Editing Image Association Information

Step 1 From the Edit Device page, click **Edit Images**.

The Edit Device Image page appears.

Step 2 Edit image and configuration information as required.

Step 3 To revert to the previous state, click **Cancel**.

Step 4 To complete this task, click **Finish**.

Resynchronizing Devices

If the password of a device becomes corrupted so that there is a mismatch between the device and the corresponding password information help in the directory, you can resynchronize the device with the Cisco Configuration Engine by using the Resync Device function.

Step 1 From the Devices Functional Overview page, click **Resync Device**.

Step 2 From the Resync Device page, click on the icon for the device you want to re-synchronize.

Note PIX devices will not be visible on this page.

Step 3 In the confirmation window that appears, click **Ok**.

Cloning Devices

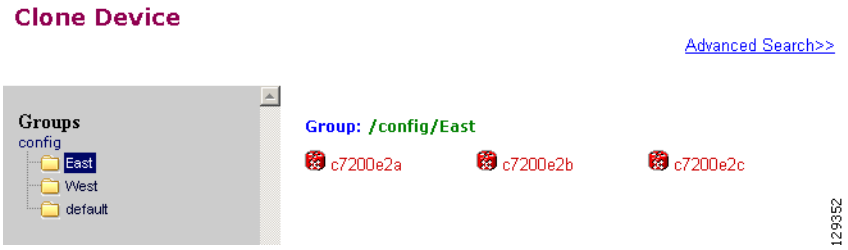
Step 1 From the Devices Functional Overview page, click **Clone Device**.

The Groups list appears.

Step 2 From the Groups list, select the group that holds the device you want to clone.

The Clone Device list appears (see Figure 3-36).

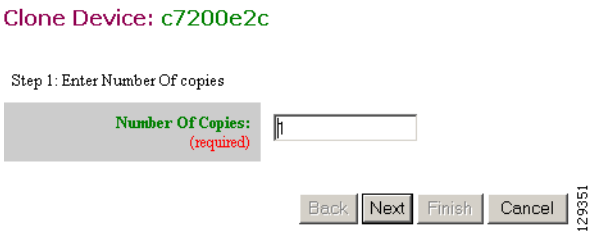
Figure 3-36 Clone Device List



Step 3 Select a device to clone.

The Step 1 page appears (see Figure 3-37).

Figure 3-37 Clone Device > Number of Copies



Step 4 Determine the number of copies, then click **Next**.

The Step 2 page appears (see Figure 3-38).

Figure 3-38 Clone Device > Name and IDs

Clone Device: c7200e2c

Step 2: Create 1 copies of c7200e2c using:

	Prefix	Suffix
Device Name	copyOf	1
Event ID	copyOf	1
Config ID	copyOf	1
Image ID	copyOf	1

Also Clone:

☑	SubDevice(s)	SubDevice Name *Prefix*	copyOf
		SubDevice ID *Prefix*	copyOf
☑	Image(s)		

Back Next Finish Cancel

129350

Step 5 Enter prefix and suffix for each device copy, then click **Next**.

The Step 3 page appears (see Figure 3-39).

Figure 3-39 Clone Device > Review Parameters

Clone Device: c7200e2c

Step 3: Review parameters

The following Devices will be created:

Device Names	Event Ids	Config Ids	Image Ids
copyOfc7200e2c1	copyOfc7200e2c1	copyOfc7200e2c1	copyOfc7200e2c1

The above devices will be created with the following attributes:

ImageRefList	C7200-IS-MZ
Template	DemoRouter.cfgtpl
ActivationTemplate	DemoRouter.cfgtpl
IOSsubdevices	card2b
Group	ou=East,ou=config,ou=CNSApplications,ou=techdoc,o=cisco,c=us
AdminDevType	generic_device

Back Next Finish Cancel

129328

Step 6 Review the parameters you set for this clone.

Step 7 If you want to make changes, click **Back**.

Step 8 To finish this task, click **Finish**.

Deleting Devices

Step 1 From the Devices Functional Overview page, click **Delete Device**.

The Groups list appears.

Step 2 From the Groups list, select the group that holds the device you want to delete.

The device list appears.

Step 3 Click the check box for the device(s) you want to delete.

Step 4 Click **Submit**.

A list of devices selected for deletion appears.

Step 5 To continue, click **Delete**.

Updating Device Configurations and Images

To send an updated version of the configuration or a new image to a device, from the Devices Functional Overview page, click **Update Device**. The Update Device Functional Overview page appears showing:

- Update Configuration
- Update Image
- Customize

Updating Device Configurations

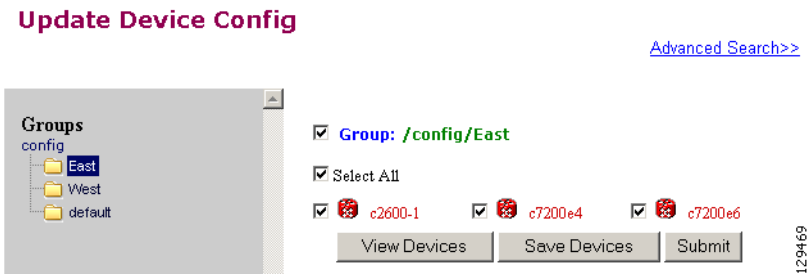
Step 1 From the Update Devices Functional Overview page, click **Update Config**.

The Groups list appears.

Step 2 From the Groups list, select the group that holds the device you want to update.

Step 3 Click the check box next to the icon for the device(s) you want to update (see Figure 3-40).

Figure 3-40 Update Config Group/Device Selection Page

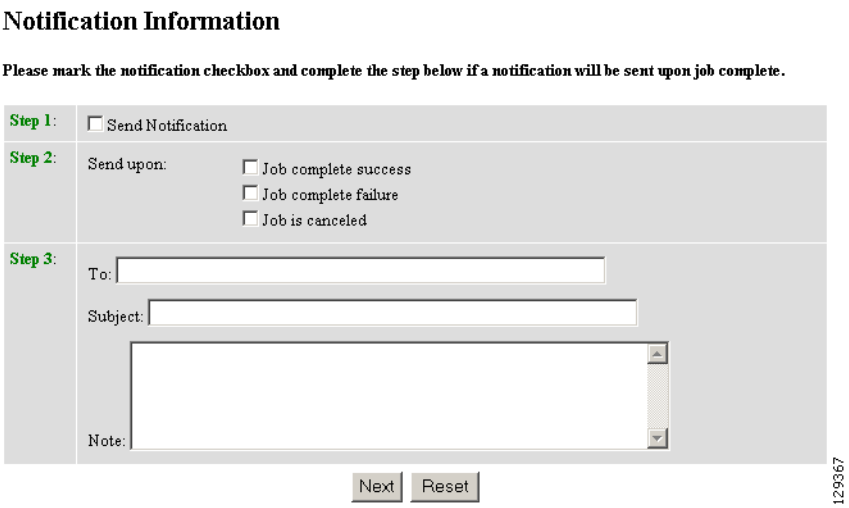


Note PIX devices will not be visible on this page.

Step 4 Click **Submit**.

The update notification page appears (see Figure 3-41).

Figure 3-41 Update Configuration Notification Information

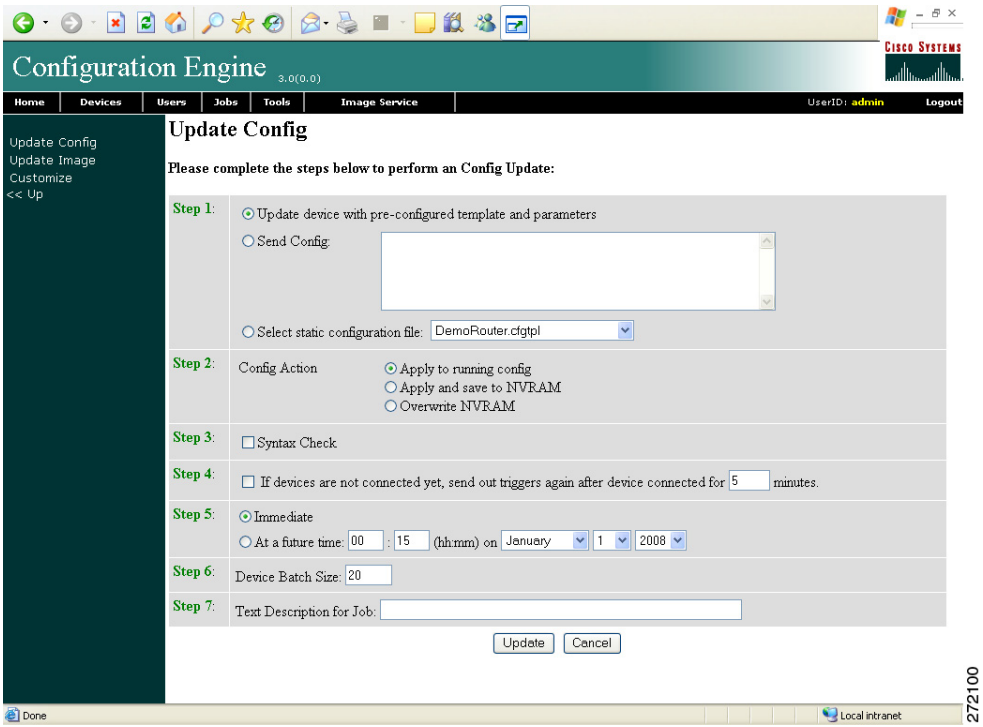


Step 5 If you want an email notification sent when the update job completes, fill in the information on this page, then click **Next**.

Note This page is optional. You can skip to the next page by clicking **Next**.

The update task dialog box appears (see Figure 3-42).

Figure 3-42 Update Task



Step 6 For Step 1, select the source of the configuration.

Step 7 For Step 2, choose the **Config Action** task you require.

- Apply to running config – applies the configuration to the current running configuration.
- Apply and save to NVRAM – applies the configuration without causing it to persist in NVRAM.
- Overwrite NVRAM – applies the change and causes it to persists in NVRAM.

Step 8 For Step 3, if required, check the **Syntax Check** check box.

Step 9 For Step 4, if devices are not connected, check this check box to send out triggers.

Step 10 For Step 5, select the date and time to send the configuration update.

Step 11 For Step 6, determine the batch size.

Tip The max batch size for IMGW should be set at 25.

Step 12 For Step 7, if applicable, enter a description for this update job.

Step 13 Click **Update**.

Updating Device Images

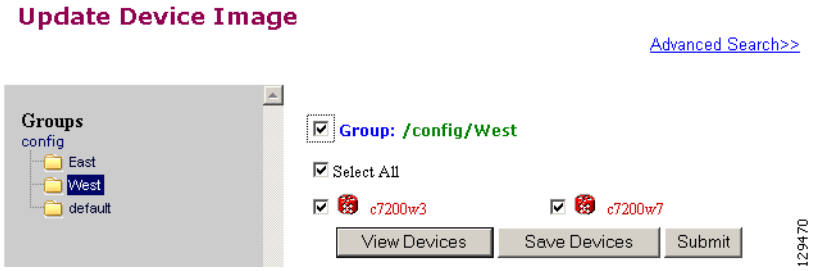
Step 1 From the Update Device Functional Overview page, click **Update Image**.

The Groups list appears.

Step 2 From the Groups list, select the group that holds the device you want to update.

Step 3 Click the check box next to the icon for the device(s) you want to update (see Figure 3-43).

Figure 3-43 Update Image Group/Device Selection Page



Note PIX/ASA devices will not be visible on this page.

Step 4 Click **Submit**.

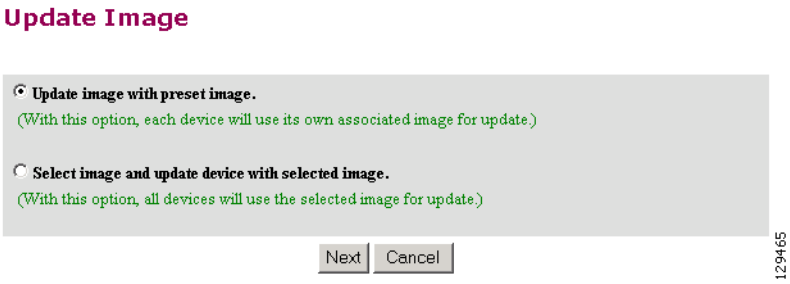
The update notification page appears (see Figure 3-41).

Step 5 If you want a notification sent when the update job completes, fill in the information on this page, then click **Next**.

Note This page is optional. You can skip to the next page by clicking **Next**.

The Update Image page appears (see Figure 3-44).

Figure 3-44 Image Selection Page

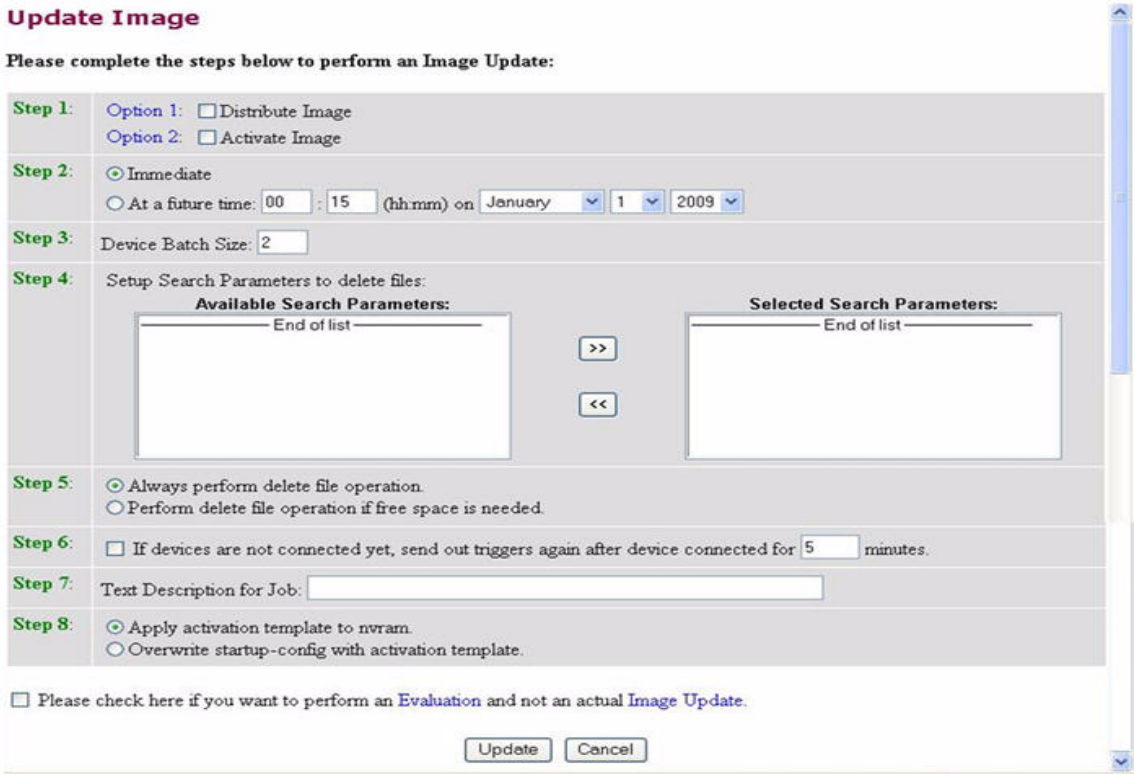


Step 6 Select the image you want to use for updates, then click **Next**.

If you select to update the device by selecting an image other than its present image, the next page gives you a list of images from which to select.

The Update Image worksheet appears (see Figure 3-45).

Figure 3-45 Update Image Worksheet



Step 7 To distribute the image, click the check box for **Distribute Image**.

Step 8 To activate the image, click the check box for **Activate Image**.

Tip All three agents (event, partial config, and image) must be running on the device for the activation process to succeed.

Note For the image to become active on the device, you must have a Configuration Control template associated with this device that contains the CLI commands for image activation (see "Configuration Control Templates" section on page 12-127).

Step 9 To update the image immediately, click the radio button for **Immediate**.

Step 10 To update the image at a specified time in the future, click the radio button for **At a future time**:

a. Enter a time value.

b. Enter a date value.

Step 11 Set the **Device Batch Size**.

This is the number of concurrent image updates. This feature allows you to limit the number of concurrent requests to a server. When one batch of image update requests has been satisfied, the next batch starts.

Tip The max batch size for IMGW should be set at 25. And for HTTP only (no event agent) mode, the batch size must be same as the number devices in the submitted job.

Note If you are running a device image update session to a mix of IMGW and agent devices, the effective device batch size limit for IMGW devices—concurrent Telnet session limit—is equal to the value (default = 25) set for this attribute in the **Setup** program (see *Cisco Configuration Engine Installation and Configuration Guide*).

Step 12 If applicable, enter a text description of the job.

Step 13 To perform an evaluation rather than an actual update, click the check box at the bottom of this pane.

Step 14 To continue, complete the steps called for, then click **Update.**

The Update Image Status page appears (see Figure 3-46). You can use this Job ID to perform job-related tasks (see Chapter 5, "Configuration and Image Update Jobs Manager").

Figure 3-46 Job ID for Update Image

Update Image Status

Device Name	Distributed Image(s)	Activated Image(s)
Device2	image3 image2	image2

Your request has been assigned the job id: 1062710890226

101509

Customize Job Template

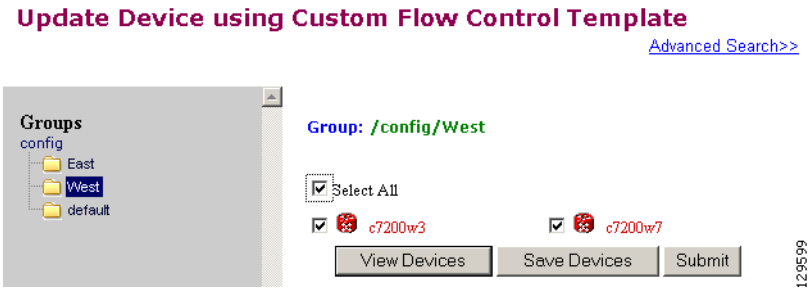
Step 1 From the Update Device Functional Overview page, click **Customize**.

The Groups list appears.

Step 2 From the Groups list, select the group that holds the device you want to update.

Step 3 Click the check box next to the icon for the device(s) you want to update (see Figure 3-47).

Figure 3-47 Custom Flow Control Device Update Selection Page

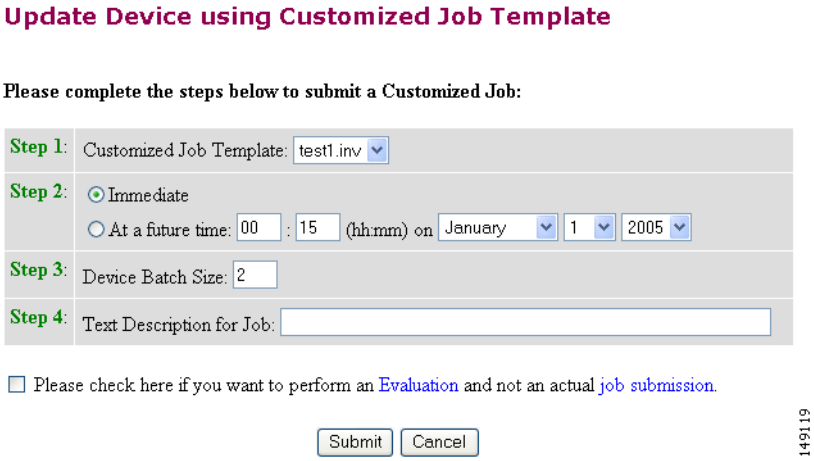


Note PIX devices will not be visible on this page.

Step 4 Click **Submit**.

The Update Device using Customized Job Template appears (see Figure 3-48).

Figure 3-48 Customized Job Template Form



Step 5 Complete the Customized Job Template form, then click **Submit**.

The next page shows the Job ID for this update task.

Step 6 To check the status of this job go to **Jobs > Query Jobs**, then click on the Job ID for this Job.

Configuration Control Template

To restart a device with a new image, you must issue the CLI commands that you would normally enter from the device console to activate a new image.

For example, if you want to restart a Cisco 3600 Series router with an image named *3600.image*, from the device console, you would issue the following CLI commands:

no boot system
boot system flash:3600.image

you must provide the device with a Configuration Control template that contains the required CLI commands for image activation.

If you do not have such a template, see "Adding a Template" section on page 12-138. Also, you must associate this Configuration Control template with the particular device (see "Adding Devices" section on page 3-31).

The content of the Configuration Control template for image activation should contain the CLI commands that you would normally enter from the device console to activate a new image on the device.

Working with Subdevices

A subdevice is a configuration object for network modules in a modular router. When working with subdevices, it is very important to pick the correct type of interface card or module.

Note PIX Firewall devices do not have subdevices.

To work with subdevices, from the Devices Functional Overview page, click **Subdevices**.

The Subdevices Functional Overview page appears showing:

- View Subdevice
- Add Subdevice
- Edit Subdevice
- Clone Subdevice
- Delete Subdevice

Viewing Subdevices

Step 1 From the Subdevices Functional Overview page, select **View Subdevice**.

The list of subdevices appears (see Figure 3-49).

Figure 3-49 View Subdevice

Step 2 Click on the icon for the device configuration you want to view.

The Configuration for that device appears.

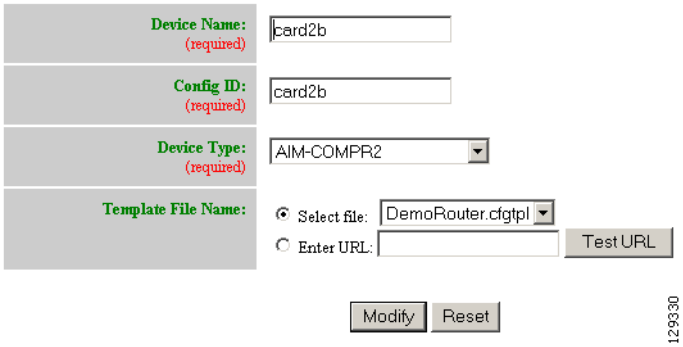
Note The subdevice configuration displayed is the configuration as it appears at the configuration server. It might not be the configuration running on the subdevice.

Adding Subdevices

Step 1 From the Subdevices Functional Overview page, click **Add Subdevice**.

The Subdevice Information page appears (see Figure 3-50).

Figure 3-50 Subdevice Information Page



Step 2 Enter a valid value (no spaces) in the **Device Name** field.

Table 3-23 shows valid values for this task.

Table 3-23 Valid Values for Add Subdevice

Attribute	Description	Valid Values
Device Name	The name used as **cn** (common name) of the device.	a-z A-Z 0-9 -(hyphen) _ (under-score) . (period)
ConfigID	Configuration ID attribute of the device.	a-z A-Z 0-9 -(hyphen) _ (under-score) . (period)

Table 3-23 Valid Values for Add Subdevice (continued)

Attribute	Description	Valid Values
Device Type		From drop-down list
Template File Name	Name of the configuration template to associate with the device.	From drop-down list, or user-defined

Step 3 Accept the default value that appears or enter another valid value (no spaces) in the **Config ID** field.

Step 4 From the **Device Type** drop-down list, choose the type of device to which this subdevice is associated.

Device type is the name of the network module as defined in the Cisco product catalog (price list).

Step 5 Choose a template file.

To use a template on your Cisco Configuration Engine:

a. Choose **Select file**.

b. Use the drop-down list to choose a template.

OR

To use an external template:

a. Choose **Enter URL**.

b. Enter the full URL for the server, directory, and filename where the template is stored. Currently, only **http** is supported.

c. To test access to the external template, click **Test URL**.

If the server is unavailable or the external template cannot be accessed, an error appears. You can still save this logical subdevice, but the template is not available until you have access to the external template.

Step 6 To clear your entries, click **Reset**.

Step 7 To add this device, click **Add**.

Editing Subdevices

Step 1 From the Subdevices Functional Overview page, click **Edit Subdevice**.

Step 2 From the Edit Subdevice page, click on the icon for the subdevice you want to edit.

The subdevice configuration appears with a menu of edit functions in the left navigation pane:

- Edit Information
- Edit Template
- Edit Parameter-Admin – Administrator-level view
- Edit Parameter-Operator – Operator-level view; used by Administrator to verify what Operator can see after Administrator has used **Edit > AttributInfo** under the Template Manager
- Edit ContactInfo

Step 3 From the left navigation pane, choose the edit function you want to use.

Editing Subdevice Information

Step 1 From the Edit Subdevice page, click **Edit Information**.

The subdevice information editor dialog box appears (see Figure 3-50).

Step 2 Modify all applicable fields.

For valid values, see Table 3-23.

Step 3 To clear your entries, click **Reset**.

Step 4 To update device information, click **Modify**.

Editing Subdevice Template

Step 1 From the Edit Subdevice left navigation pane, click **Edit Template**.

The template editor appears.

Step 2 In the **Attributes** field, click the drop-down arrow.

Step 3 Choose the attribute you want to add to the template, then click **Add**.

Step 4 Repeat Steps 2 and 3 for all attributes you want to add to the template file.

Step 5 Delete all unusable strings from the template file.

Step 6 Edit strings as necessary.

The default multi-line begin and end tags are **^C** and **^C** respectively. The delimiter for these tags are: ~ ! @ ^ & * - = |. Do not use # or %.

A multi-line test banner might be:

```
banner exec ^C
    This is a Test Banner
    1. Hi
    2. Hello
    3. Test is 1234567890
^C
```

Step 7 To save your edits, click **Save**.

Step 8 To save this version as a new template, click **Save as**.

Editing Subdevice Parameters

Step 1 From the Edit Subdevice left navigation pane, click **Edit Parameter-Admin**.

The parameters editor appears.

Note Operator-level privileges do not include access to these parameters.

Step 2 Modify parameters values as required.

Step 3 To save your edits, click **Save Parameters**.

Editing Contact Information

Step 1 From the Edit Device left navigation pane, click **Edit ContactInfo**.

The contact information appears.

Step 2 Edit all active fields as required.

Step 3 To clear your entries, click **Reset**.

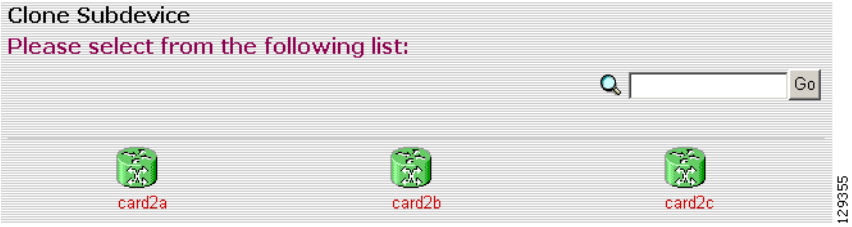
Step 4 To save your edits, click **Update**.

Cloning Subdevices

Step 1 From the Subdevices Functional Overview page, click **Clone Subdevice**.

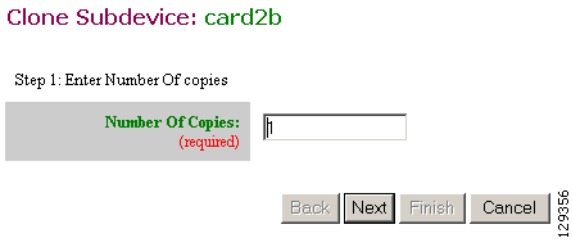
The Subdevice list appears (see Figure 3-51).

Figure 3-51 Clone Subdevice Device List



Step 2 The Step 1 page appears (see Figure 3-52).

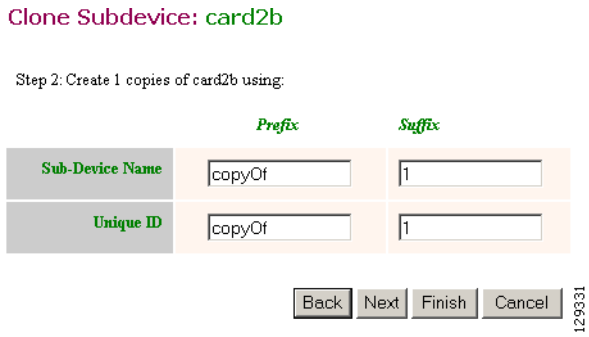
Figure 3-52 Clone Subdevice > Number of Copies



Enter the number of copies you want to make, then click **Next**.

The Step 2 page appears (see Figure 3-53).

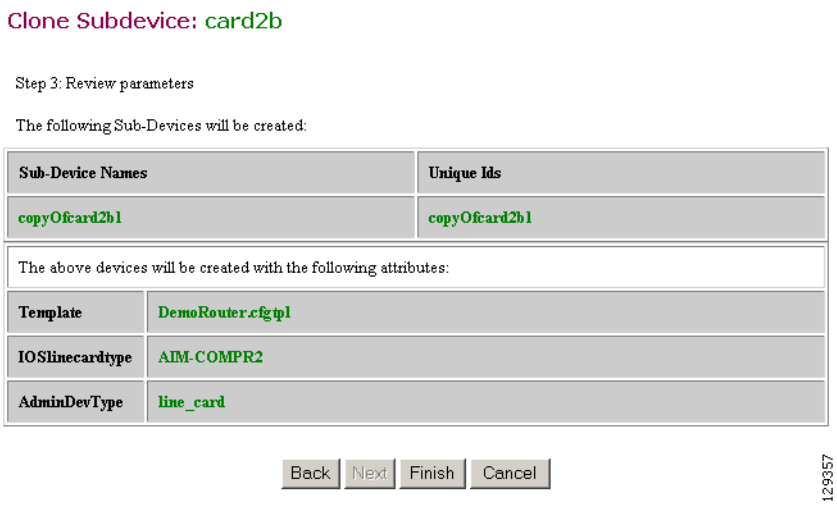
Figure 3-53 Clone Subdevice > Name and IDs



Step 3 Enter prefix and suffix for each device copy, click **Next**.

The Step 3 page appears (see Figure 3-54).

Figure 3-54 Clone Subdevice > Review Parameters



Step 4 Review the parameters you set for this clone.

Step 5 If you want to make changes, click **Back**.

Step 6 To finish this task, click **Finish**.

Deleting Subdevices

Step 1 From the Subdevices Functional Overview page, click **Delete Device**.

The Delete Subdevice page appears (see Figure 3-55).

Figure 3-55 Select Subdevices to Delete

Step 2 Check to select the subdevice(s) you want to delete.

Step 3 To proceed, click **Next**.

A status page appears indicating that the subdevice has been selected for deletion (see Figure 3-56).

Figure 3-56 Delete Subdevices Confirmation

Step 4 To delete this subdevice, click **Delete**.

Querying Device Inventory

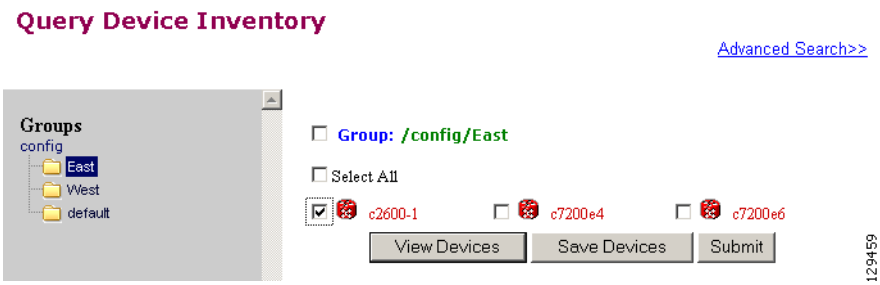
You can use the Query Device Inventory feature to get a reports from devices about:

- Running image information
- Hardware information
- File system list

Step 1 From the Devices Functional Overview page, click **Query Device Inventory**.

The Query Device Inventory screen appears.

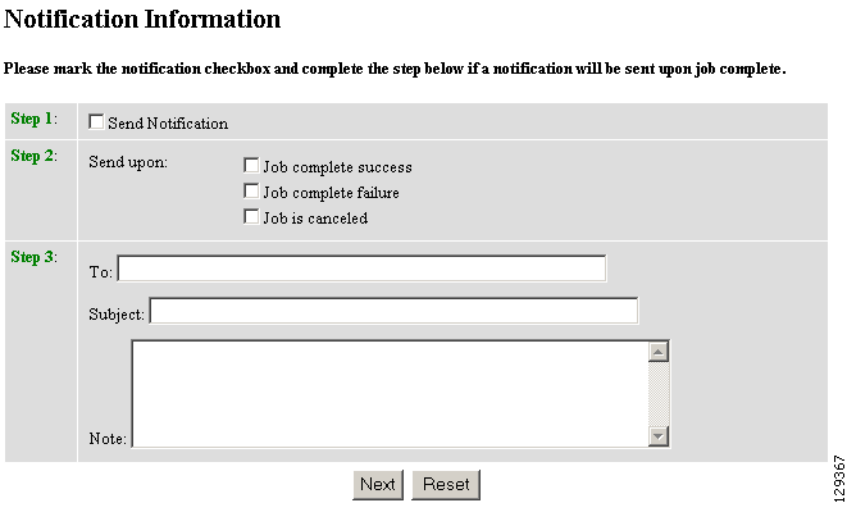
Figure 3-57 Query Device Inventory Page



Step 2 Check the device(s) for which you want to get an inventory report(s), then click **Submit**.

The Query Notification Information page appears (see Figure 3-58).

Figure 3-58 Query Notification Information Page

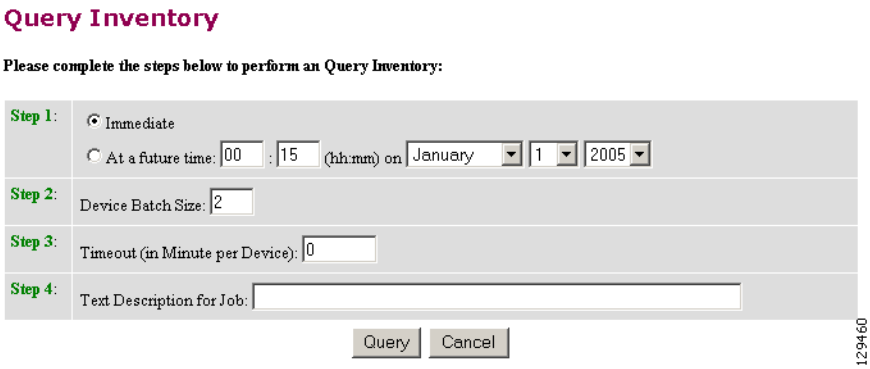


Step 3 If you want an email notification sent when the query completes, fill in the information on this page, then click **Next**.

Note This page is optional. You can continue by clicking **Next**.

The Query Attributes Page appears (see Figure 3-59).

Figure 3-59 Query Attributes Page.



Step 4 Set all applicable attributes, then click **Query**.

The query is submitted as a **Job**. A page appears indicating the job number for this query.

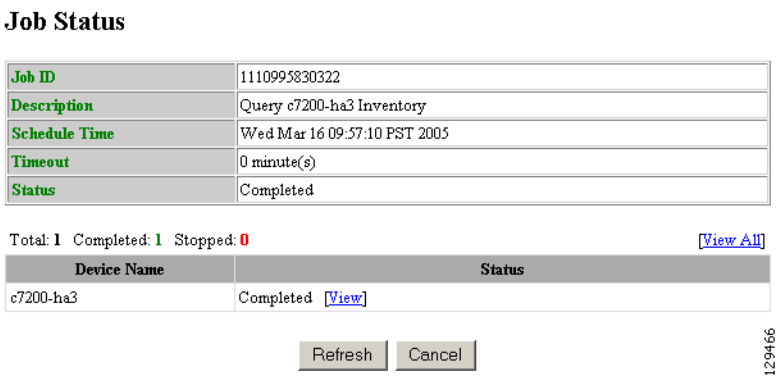
Step 5 To check the status of this job, go to **Jobs > Query Job**.

Step 6 Use the drop-down arrow to select Completed Jobs.

Step 7 For the Inventory Job you want, click either the job number or the entry in the Status column.

The Job Status page appears (see Figure 3-60).

Figure 3-60 Job Status Page



Step 8 To view the inventory report, click **View**.

Device inventory report appears (see Figure 3-61).

Figure 3-61 Sample Device Inventory Report

ImageID:c2600-1 Reported Time: 1993-03-05T22:57:37

Running Image Information					
Description (Version String)	12.2(12h)				
Image File	flash:c2600-ik8o3s-mz.122-12h	Image MD5			
Config Variable		Config Reg		Config Reg Next Boot	
Boot Variable		BootIdr Variable		Return To ROM Reason	reload
Return To ROM Time	2003-11-04T00:00:00	Started At	2003-11-04T00:00:00		

Hardware Information					
Vendor	cisco	Platform Name	2611	Hardware Revision	0x202
Processor Type		Main Mem Size	30649288	IO Mem Size	4194312
Hardware Serial #	JAB03170532	MidPlane Version			
Processor Rev					
Hardware Rework					

File System List	
	[FileSys name=[nvram:], type=[nvram], size=[29688], freespace=[26473], readable=[1],
	writeable=[1]
	Directory 0: name=[/], fullname=[nvram:/], size=[29688], readflag=[1],
	writeflag=[1], owner=[], modDate=[1969-12-31T00:00:00]
	File 0 under Directory[/]: name=[startup-config],
	fullname=[nvram:/startup-config],
	size=[1110], readflag=[1], writeflag=[1], owner=[],
	modDate=[1969-12-31T00:00:00],

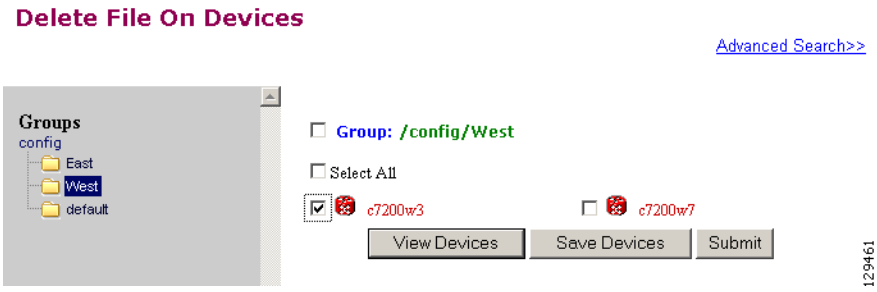
101485

Delete Files on Device

Step 1 From the Devices Functional Overview page, click **Delete Files on Device**.

The Delete File on Device page appears (see Figure 3-62).

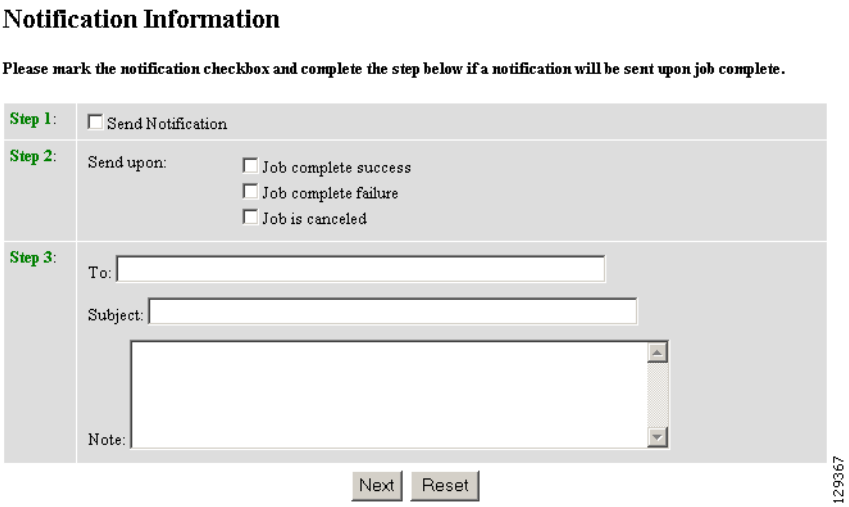
Figure 3-62 Delete Files on Device Page



Step 2 Check the device(s) on which you want to delete files, then click **Submit**.

The Delete Device Files Notification Information page appears (see Figure 3-63).

Figure 3-63 Delete Device Files Notification Information Page

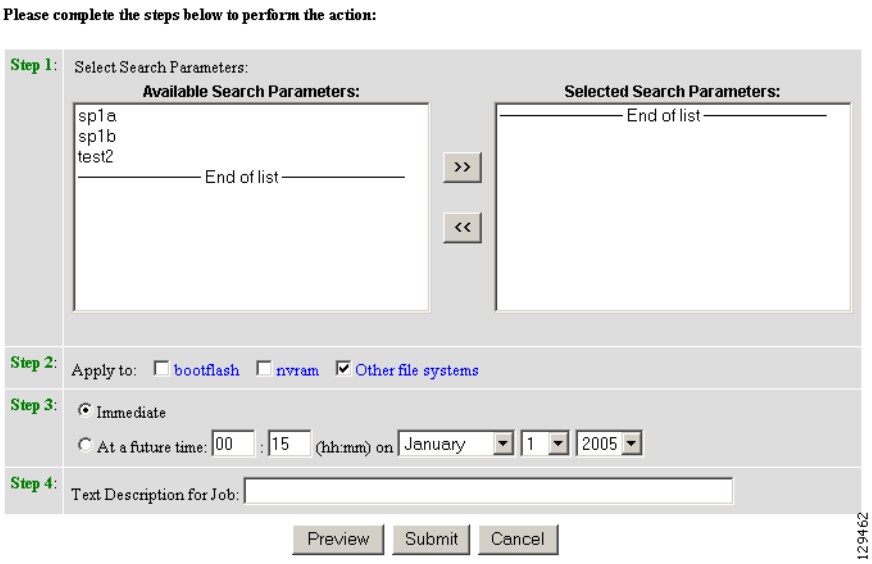


Step 3 If you want an email notification sent when the query completes, fill in the information on this page, then click **Next**.

This page is optional. You can continue by clicking **Next**.

The Delete Files parameter page appears (see Figure 3-64).

Figure 3-64 Delete Files Parameter Page



Step 4 Complete the steps on this page, then to preview, click **Preview**.

Step 5 When you are satisfied with the task parameters, click **Submit**.

Dynamic Operations

Dynamic Operations allows you to perform operations on devices that all respond to having the same attributes based on the Query used to find them.

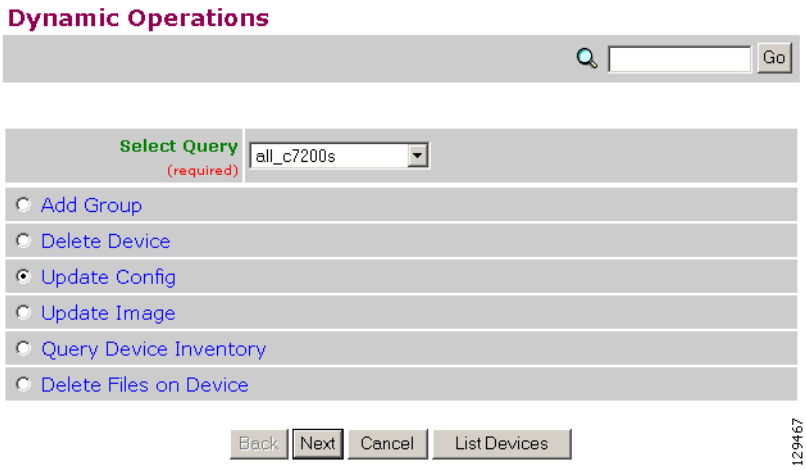
To use this feature you must have query objects available before starting Dynamic Operations. If no Queries have been created, you will see a message stating that there are no query objects available.

To create a Query, go to the "Creating Queries" section on page 8-108.

Step 1 From the Devices Functional Overview page, click **Dynamic Operations**.

The Dynamic Operations page appears (see Figure 3-65).

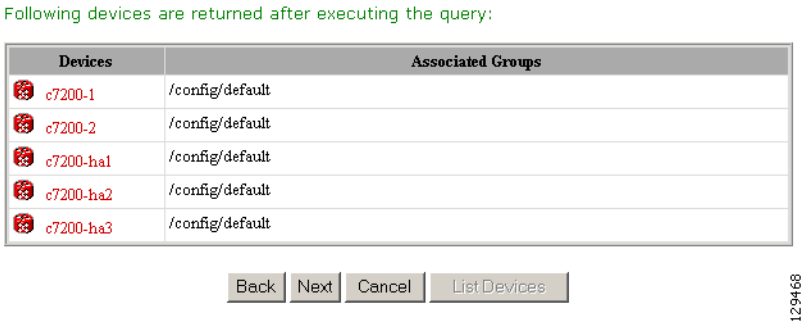
Figure 3-65 Dynamic Operations Page



Step 2 Use the down-arrow key to select the Query you want to use.

Step 3 Select the operation you want to perform on devices that respond to the Query, then click **List Devices**. The result of the Query appears (see Figure 3-66).

Figure 3-66 Devices Responding to Query



Step 4 To continue with the selected operation, click **Next**.

User Account Manager

Note User accounts can be accessed only when operating in Internal Directory mode.

To access User tasks, log in to the system (see "Logging In" section on page 2-23). Then, from the Home page, click the **Users** tab.

A functional overview of the user administration options appears showing:

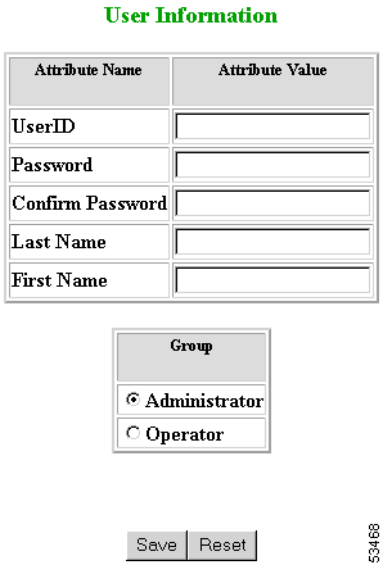
- Add User
- Edit User
- Delete User
- Change Password

Adding User Account

Step 1 From the User Administration page, click **Add User**.

The User Information dialog box appears (see Figure 4-1).

Figure 4-1 User Information



Step 2 Enter a valid value (no spaces) in the **UserID** field.

Table 4-1 shows valid values for these fields.

Table 4-1 Valid Values for Add User Account

Attribute	Description	Valid Values
UserID	ID that allows user to log in to the user interface.	a-z A-Z 0-9 -(hyphen) _ (under-score) . (period)
Password	Password	Printable characters with a length of 6 – 12
Confirm Password	Password	Printable characters with a length of 6 – 12
Last Name	Last name of registered user.	a-z A-Z 0-9 -(hyphen) _ (under-score) . (period)
First Name	First name of registered user.	a-z A-Z 0-9 -(hyphen) _ (under-score) . (period)

Step 3 Enter a password in the **Password** field.

Step 4 Confirm the password by entering it again in the **Confirm Password** field.

Step 5 Enter the user's last name in the **Last Name** field.

Step 6 Enter the user's first name in the **First Name** field.

Step 7 In the Group pane, click the radio button that classifies the privilege level (**Administrator**, **Operator**) of this user.

Step 8 To clear your entries, click **Reset**.

Step 9 To save your entries, click **Save**.

Editing User Account

Step 1 From the User Administration page, click **Edit User**.

A shows of users appears (see Figure 4-2).

Figure 4-2 User List

Step 2 From the User List, click on the icon for the user account you want to edit.

Note Administrator-level users are shown with a key icon associated with the figure icon.

The User Information page appears (see Figure 4-3).

Figure 4-3 User Information

Step 3 To modify the user ID, enter a valid value (no spaces) in the **UserID** field.

Table 4-2 shows valid values for these fields.

Table 4-2 Valid Values for User Information

Attribute	Description	Valid Values
UserID	ID that allows user to log in to the user interface.	Information only
Password	Password	Printable characters with a length of 6 – 12
Confirm Password	Password	Printable characters with a length of 6 – 12
Group	Administrator or Operator level	Radio Button

Step 4 To modify the user's last name, edit the **Last Name** field.

Step 5 To modify the user's first name, edit the **First Name** field.

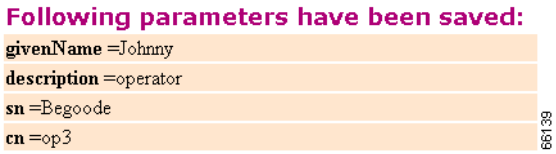
Step 6 To modify the user group status, click the appropriate radio button in the **Group** pane.

Step 7 To clear your entries, click **Reset**.

Step 8 To save your entries, click **Save**.

User information update status appears (see Figure 4-4).

Figure 4-4 User Information Update Status



Deleting User Account

Step 1 From the User Administration page, click **Delete User**.

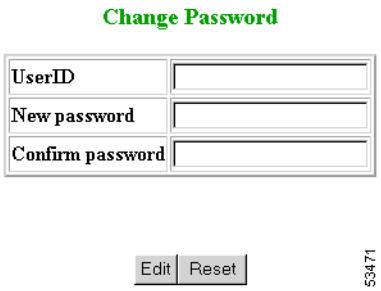
Step 2 From the user list (see Figure 4-2), click on the icon for the user account you want to delete.

Changing User Password

Step 1 From the User Administration page, click **Change Password**.

The Change Password dialog box (see Figure 4-5) appears.

Figure 4-5 Change Password



Step 2 Enter the **UserID** for the user account password you want to change or reset.

Table 4-3 shows valid values for these fields.

Table 4-3 Valid Values for Change Password by Administrator

Attribute	Description	Valid Values
UserID	ID that allows user to log in to the user interface.	a-z A-Z 0-9 -(hyphen) _ (under-score) . (period)
Password	Password	Printable characters with a length of 6 – 12
Confirm Password	Password	Printable characters with a length of 6 – 12

Step 3 Enter the new password in the **New password** field.

Step 4 Enter the new password again in the **Confirm password** field.

Step 5 To clear your entries, click **Reset**.

Step 6 To save the new password, click **Edit**.

Changing Account Privilege Level

Step 1 From the User Administration page, click **Edit User**.

Step 2 Choose the user in question from the user list (see Figure 4-2).

The User Information page appears (see Figure 4-6).

Figure 4-6 User Information

Step 3 In the Group pane, click the radio button that classifies the privilege level (Administrator, Operator) of this user.

Step 4 To clear your entries, click **Reset**.

Step 5 To save your entries, click **Save**.

Configuration and Image Update Jobs Manager

To access tasks for managing configuration and image update Jobs, log into the system (see "Logging In" section on page 2-23). Then, from the Home page, click the **Jobs** tab.

The Jobs Functional Overview page appears showing:

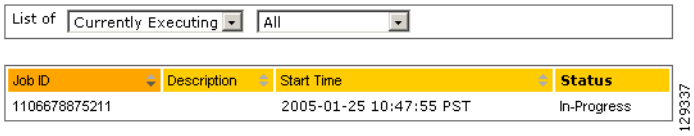
- Query Job
- Cancel/Stop Job
- Restart Job
- Delete Completed Job

Querying Jobs

Step 1 From the Jobs Functional Overview page, click **Query Job**.

The Query Job page appears (see Figure 5-1).

Figure 5-1 Query Job



Step 2 Use the drop-down arrow in the left menu to select available list of jobs:

- Currently Executing
- Stopped
- Completed

Step 3 Use the drop-down arrow in the right menu to select the type of listing:

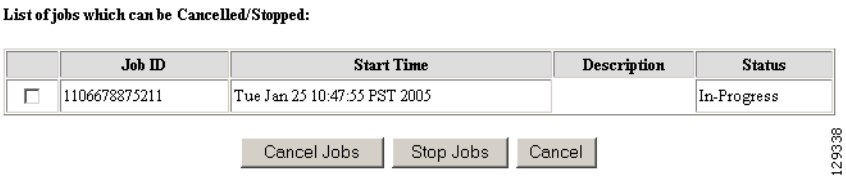
- All
- Image Jobs
- Config Jobs
- Delete Files Jobs
- Query Inventory Jobs

Canceling or Stopping Jobs

Step 1 From the Jobs Functional Overview page, click **Cancel/Stop Job**.

The Cancel/Stop Job page appears (see Figure 5-2).

Figure 5-2 Cancel/Stop Job



Step 2 Check to select the Job you want to cancel or stop, then click **Cancel Jobs**, or **Stop Jobs**.

Restarting Jobs

Step 1 From the Jobs Functional Overview page, click **Restart Job**.

The Restart Job page appears (see Figure 5-3).

Figure 5-3 Restart Job

Step 2 Check to select the Job you want to restart, then click **Restart Jobs**.

Step 3 After the Cisco Configuration Engine restarts, the prior in-progress job will be in stopped state. Restarting such a job will make the job invalid.

Deleting Completed Jobs

Step 1 From the Jobs Functional Overview page, click **Delete Completed Jobs**.

The Delete Completed Jobs page appears (see Figure 5-4).

Figure 5-4 Completed Jobs List

Delete Completed Job

List of Completed jobs:

Select All

	Job ID	Start Time	Description	Status
	1107916464334	Tue Feb 08 18:34:24 PST 2005		Completed
	1107917966260	Tue Feb 08 18:59:26 PST 2005		Completed
	1107921621739	Tue Feb 08 20:00:21 PST 2005		Completed
	1107921920495	Tue Feb 08 17:05:19 PST 2005	Submit through WEB SERVICE API @ 1107911119786	Completed
	1107975382765	Wed Feb 09 10:56:22 PST 2005		Completed

Delete Jobs Cancel

129448

Step 2 Check to select the completed jobs you want to delete, then click **Delete Jobs**.

Groups

To access Group management tasks, log into the system (see Logging In, page 2-23). Then, from the Home page, click the **Tools** tab. The Tools page appears.

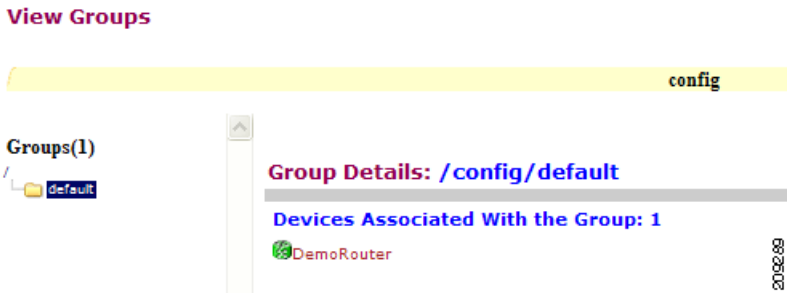
From the Tools page, click **Group Mgr**. The Group Management page appears showing:

- View Groups
- Create Group
- Edit Group
- Clone Group
- Move Group
- Delete Groups
- Create Group Using Search

Viewing Groups

From the Group Management page, click **View Groups**. The View Groups page appears (see Figure 6-1).

Figure 6-1 View Groups Page

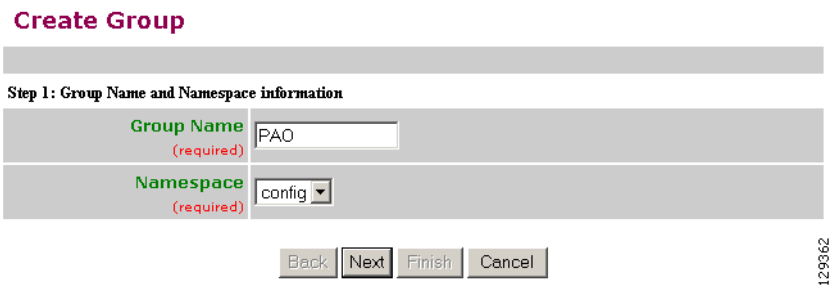


Creating Groups

Step 1 From the Group Management page, click **Create Groups**.

The Create Groups page appears (see Figure 6-2).

Figure 6-2 Create Group

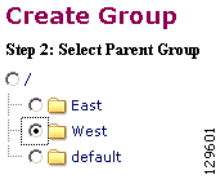


Step 2 Enter the group name.

Step 3 Use the drop-down arrow to select a namespace value (only **config** available), then click **Next**.

The Select Parent Group page appears (see Figure 6-3).

Figure 6-3 Select Parent Group Page



Step 4 Click the radio button(s) to select the parent group with which you want the new group to associated, then click **Next**.

The Select Member Devices page appears (see Figure 6-4).

Figure 6-4 Select Member Devices Page

Step 5 Check to select the devices you want to be in this group, then click **Finish**.

Editing Groups

Step 1 From the Group Management page, click **Edit Group**.

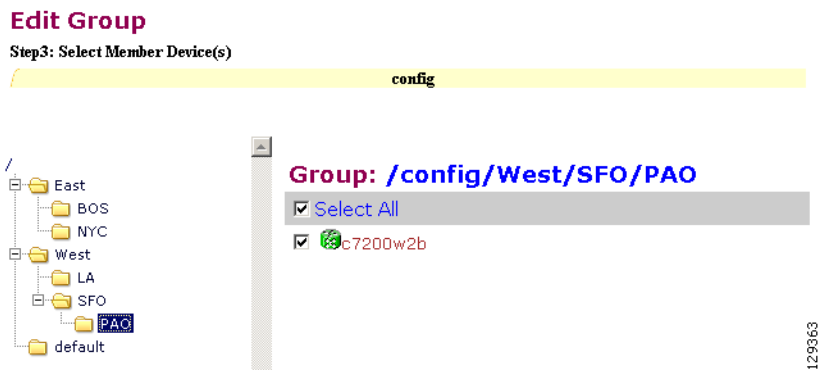
The Group list appears.

Step 2 Click the radio button to select a group to edit, then click **Next**.

The Rename Group page appears.

Step 3 Rename group, if applicable, then click **Finish** to complete the task, or click **Next** to continue (see Figure 6-5).

Figure 6-5 Edit Group Members



Step 4 Click the Group you are editing to bring up its members.

Step 5 Modify the members in this group by using the check box next to each member, then click **Finish**.

Cloning Groups

Step 1 From the Group Management page, click **Clone Group**.

The Group list appears.

Step 2 Select a group to clone.

Step 3 Select parent group.

Step 4 Enter new group name. After cloning a group, the devices in the original group will exist in the cloned group.

Moving Groups

Step 1 From the Group Management page, click **Move Group**.

The Group list appears.

Step 2 Select a group to move.

Step 3 Select parent group.

Deleting Groups

Step 1 From the Group Management page, click **Delete Groups**.

The Group list appears.

Step 2 Check to select the group(s) you want to delete.

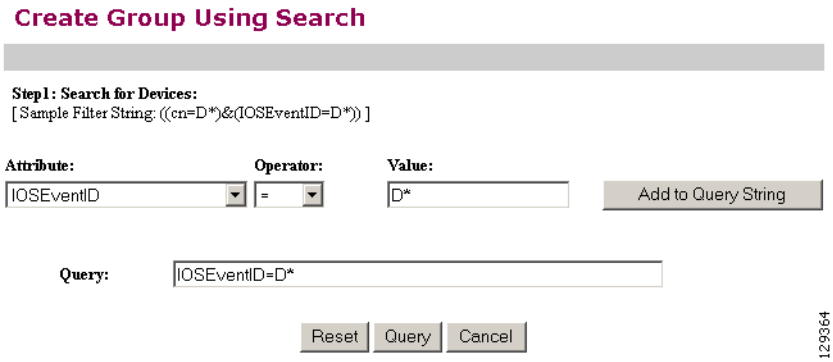
Note When you delete a group, the devices associated with that group will not be deleted.

Creating Groups Using Search

Step 1 From the Group Management page, click **Create Group Using Search**.

The search for devices page appears (see Figure 6-6).

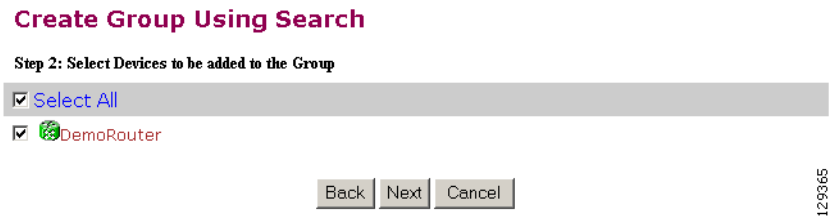
Figure 6-6 Search for Devices



Step 2 Enter the appropriate arguments for the search, then click **Query**.

Any devices found appear on the next page (see Figure 6-7).

Figure 6-7 Select Devices to Add to Group



Step 3 Check to select the devices you want to become members of this new group, then click **Next**.

The next page (see Figure 6-8) gives you the choice to add a new group, or just add the devices found to an existing group.

Figure 6-8 Name Group and Namespace

Step 4 Enter group name.

Step 5 Use the drop-down arrow to select a namespace value, then click **Next**.

The group list page appears.

Step 6 Select group parent, then click **Finish**.

Namespace Manager

The Namespace Manager provides a GUI for managing the system namespace known as "config," which contains the set of Cisco standardized events, such as *com.cisco.cns.mgmt.config.load*, etc. By default, each event defines a mapping to itself for both the publish and subscribe mapping.

If you are using the *Cisco Configuration Engine Software Development Kit API Reference and Programmer Guide* to develop your own application, you are free to redefine the map according to your application needs. Additional application-specific namespace values can be defined by means of the Cisco Configuration Engine SDK.

Note Cisco Configuration Engine supports multiple namespaces and their respective mappings by means of the Cisco Configuration Engine GUI.

The system namespace is guaranteed to return a mapping even for undefined events; in which case, the input map is returned as the output map. This is a requirement for supporting future devices which might depend on new events that are not currently defined.

To access Namespace management tasks, log into the system (see "Logging In" section on page 2-23). Then, from the Home page, click the **Tools** tab. The Tools page appears.

From the Tools page, click **Namespace Mgr**. The Namespace Management page appears showing:

- View Events
- Add Events
- Edit Events
- Delete Events

Viewing Events

From the Namespace Manager main page, click **View Events**. The events list for the current application (config) appears (see Figure 7-1).

Figure 7-1 Events List

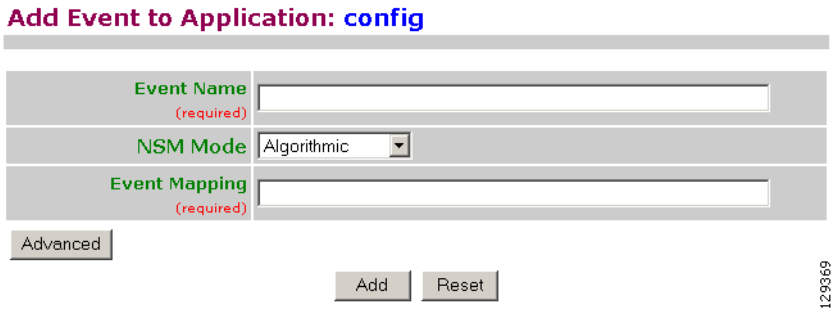
Adding Events

The events list for the current application (config) appears (see Figure 7-1).

Step 1 From the Namespace Manager main page, click **Add Events**.

The Event information page appears (see Figure 7-2).

Figure 7-2 Event Information Page



Step 2 Enter an Event name.

Step 3 Use the drop-down arrow to select the NSM Mode.

- Algorithmic – Mapped events qualified with group name or device name are returned from NSM. This is the preferred mode for all users. It allows you to provision the selected group(s) of device(s).
- Non-Algorithmic – Mapped events are returned from NSM without group name or device name. You are forced to provision all device(s).

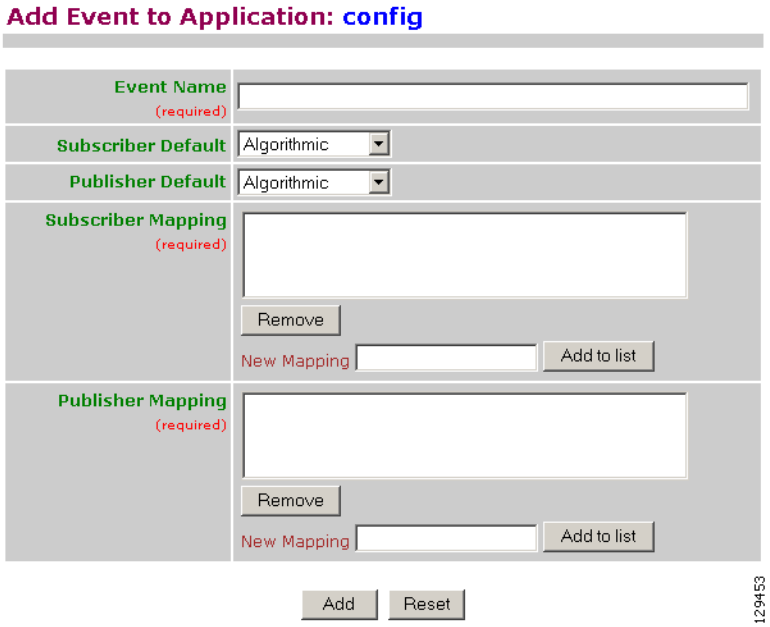
Step 4 Enter a valid Event Mapping.

For example: **cisco.mgmt.cns.exec.reload**

Step 5 To define separate parameters for Subscriber Mapping and Publisher Mapping, click **Advanced**.

The advanced event information page appears (see Figure 7-3).

Figure 7-3 Advanced Event Information Page



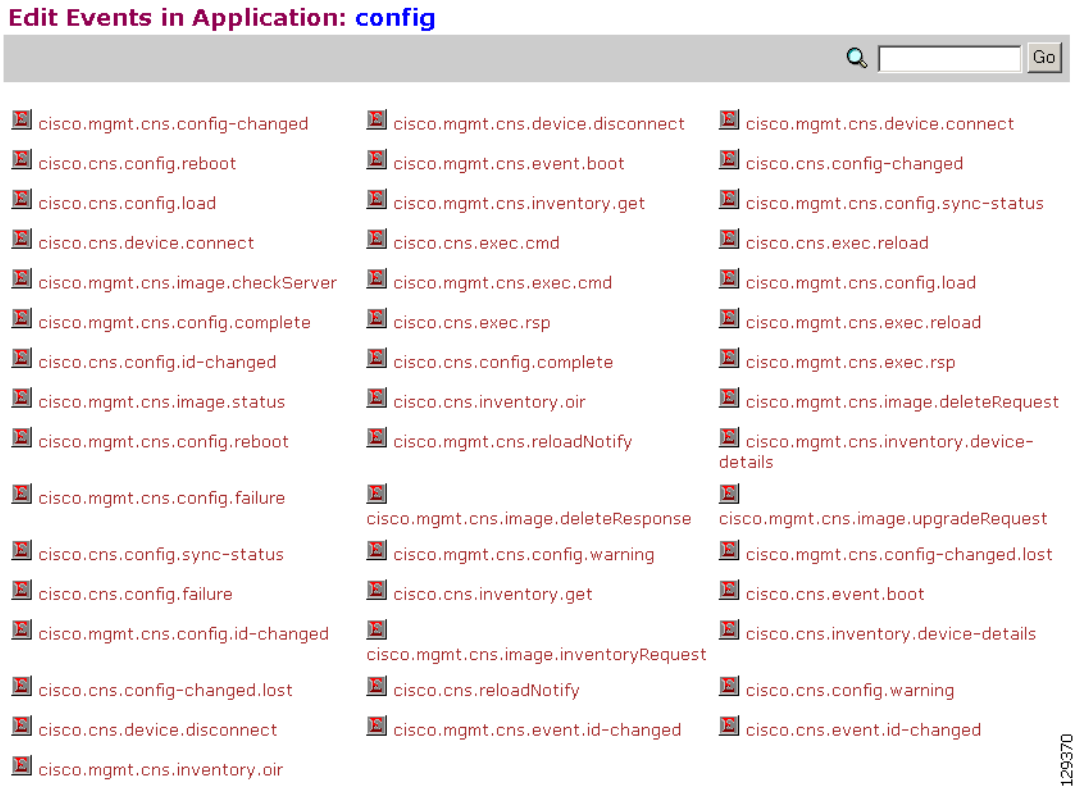
Step 6 Enter information in the appropriate fields, then click **Add**.

Editing Events

Step 1 From the Namespace Manager main page, click **Edit Events**.

The Event information page appears (see Figure 7-4).

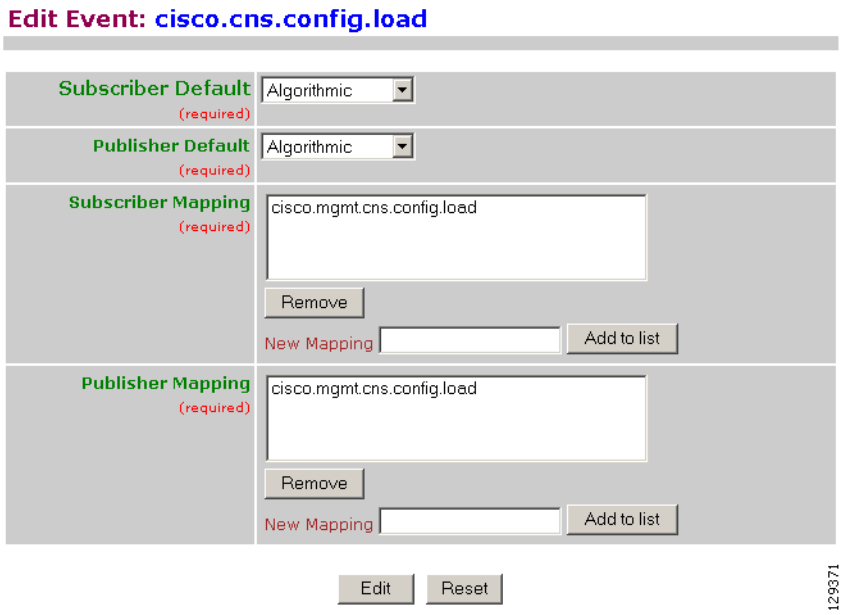
Figure 7-4 Event List to Edit



Step 2 Click on the Event you want to edit.

The Edit Event parameters page appears (see Figure 7-5).

Figure 7-5 Edit Event Parameters



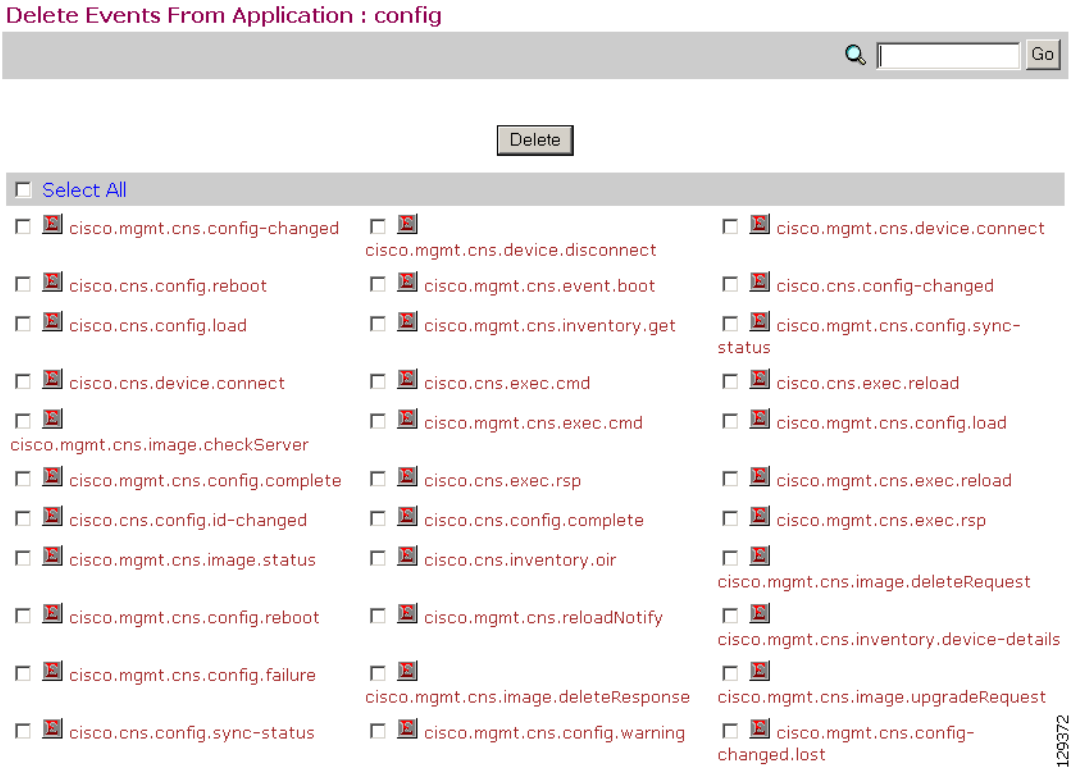
Step 3 Modify all applicable fields, then click **Edit**.

Deleting Events

Step 1 From the Namespace Manager main page, click **Delete Events**.

The Delete Event list page appears (see Figure 7-6).

Figure 7-6 Event List for Deleting Events



Step 2 Check to select the Event(s) you want to delete, then click **Delete**.

A confirmation box appears.

Step 3 To Delete the selected Event(s), click **OK**.

Query Manager

To access Query management tasks, log into the system (see "Logging In" section on page 2-23). Then, from the Home page, click the **Tools** tab. The Tools page appears.

From the Tools page, click **Query Mgr**. The Query Manager Functional Overview page appears showing:

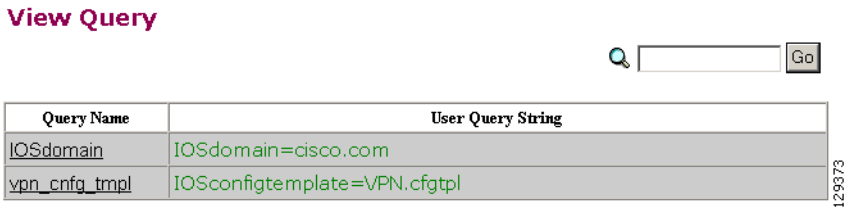
- View Query
- Create Query
- Edit Query
- Delete Query

Viewing Queries

Step 1 From the Query Manager Functional Overview page, click **View Query**.

The View Queries page appears (see Figure 8-1).

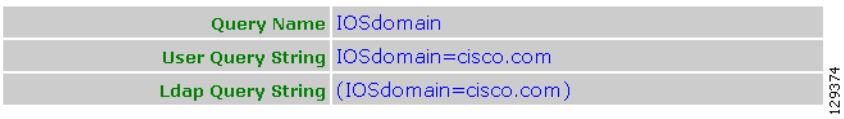
Figure 8-1 View Queries Page



Step 2 Click on the Query Name for which you want to view details.

The Query Details page appears (see Figure 8-2).

Figure 8-2 Query Details

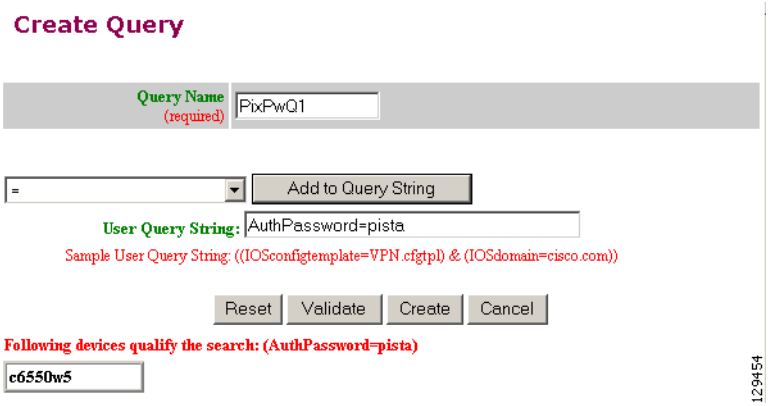


Creating Queries

Step 1 From the Query Manager Functional Overview page, click **Create Query**.

The Create Query page appears (see Figure 8-3).

Figure 8-3 Create Query Page



Step 2 Enter a Query Name.

Step 3 Use the drop-down arrow to select Operators and Attributes with which to build a Query String, then for each successive click **Add to Query String**.

Each time you click **Add to Query String**, that portion of the argument is added to the query string.

Step 4 If required, enter the remainder of the argument in the User Query string field.

Step 5 To validate this query before you create it, click **Validate**.

The Query returns a result.

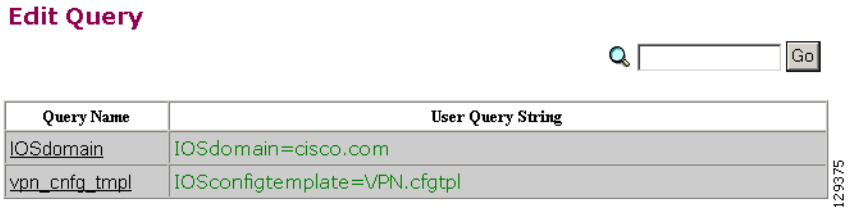
Step 6 To create this query, click **Create**.

Editing Queries

Step 1 From the Query Manager Functional Overview page, click **Edit Query**.

The Edit Query page appears (see Figure 8-4).

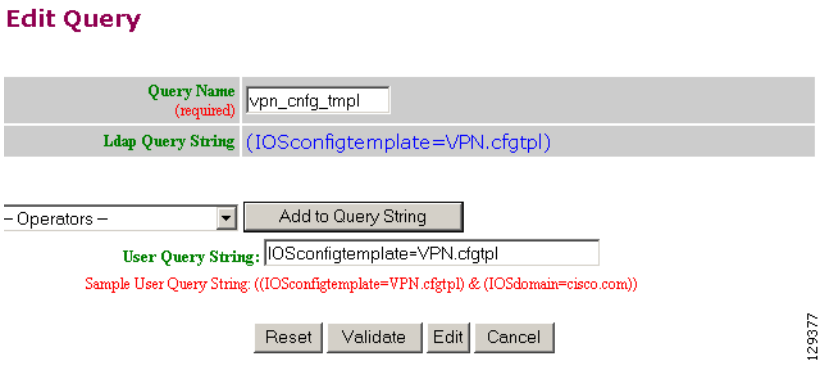
Figure 8-4 Edit Query Page



Step 2 Click on the Query Name you want to edit.

The Edit Query Attributes page appears (see Figure 8-5).

Figure 8-5 Edit Query Attributes Page



Step 3 Modify all applicable fields:

a. Use the drop-down arrow to select Operators and Attributes with which to build a Query String, then for each successive click **Add to Query String**.

Each time you click **Add to Query String**, that portion of the argument is added to the query string.

b. If required, enter the remainder of the argument in the User Query string field.

c. To validate this query before you create it, click **Validate**.

The Query returns a result.

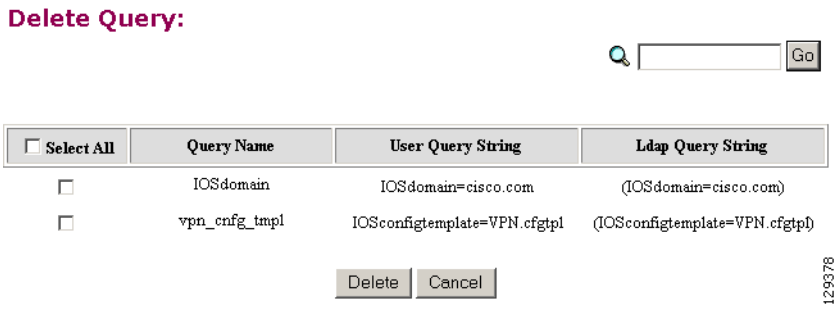
Step 4 To save your changes to this query, click **Edit**.

Deleting Queries

Step 1 From the Query Manager Functional Overview page, click **Delete Query**.

The Delete Query page appears (see Figure 8-6).

Figure 8-6 Delete Query Page



Step 2 Check to select the Query you want to delete, then click **Delete**.

Data Manager

To access Data management tasks, log into the system (see "Logging In" section on page 2-23). Then, from the Home page, click the **Tools** tab. The Tools page appears.

From the Tools page, click **Data Manager**. The Data Manager page appears. The Data Manager functions include:

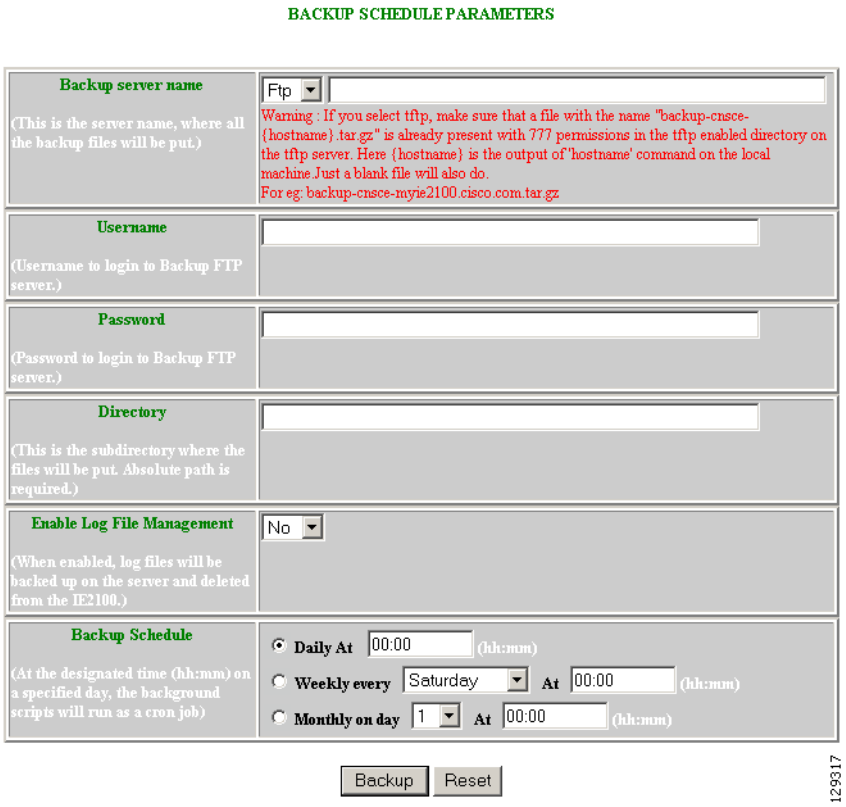
- Schedule Backup
- Update Product List
- Manage Disk Space

Scheduling Data Backup

Step 1 From the Data Manager Overview page, click **ScheduleBackup**.

The backup information dialog box appears (see Figure 9-1).

Figure 9-1 Backup Schedule Parameters



Step 2 To specify where you want the backup data to be stored, enter the FTP server name in the **FTP Server Name** field.

Table 9-1 shows valid values for these fields.

Table 9-1 Valid Values for Backup Schedule Parameters

Attribute	Description	Valid Values
FTP Server name	Server name where all backup files will be put.	a-z A-Z 0-9 -(hyphen) _ (under-score) . (period)
Username	Login username for the FTP server.	a-z A-Z 0-9 -(hyphen) _ (under-score) . (period)
Password	Password for FTP server.	Printable characters with a length of 6 – 12
Directory	Subdirectory into which all backup files will be put.	Absolute path

Table 9-1 Valid Values for Backup Schedule Parameters (continued)

Attribute	Description	Valid Values
Enable Log File Management	determines whether files will be deleted from host system after backup.	From drop-down list
Backup Schedule	Date and time fields.	As required

Step 3 To specify the username to log into the FTP server, enter a valid username in the **Username** field.

Step 4 To specify the password to use to log into the FTP server, enter a valid value in the **Password** field.

Step 5 To specify the subdirectory where the data file is put, enter the absolute path in the **Directory** field.

Step 6 Choose whether to **Enable Log File Management**.

Step 7 To specify the backup schedule, complete the fields in the **Backup Schedule** pane.

Note The time base for the host system should be set to Coordinated Universal Time (UTC).

Step 8 To cancel the backup operation, click **Cancel**.

Step 9 To start the backup operation, click **Backup**.

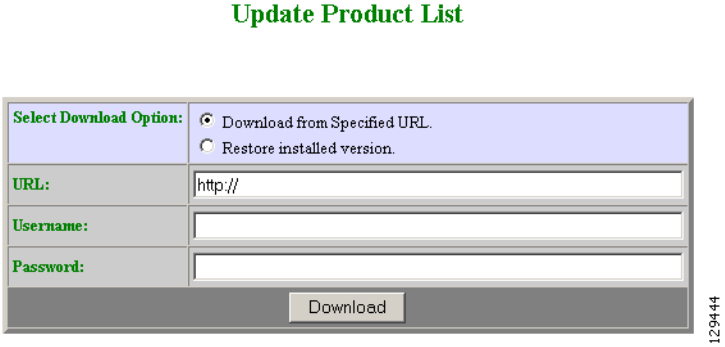
Updating Product List

The product list is a mapping between product name of the network modules as specified in the pricing list and the numeric identification number stored in EPROM. As new products are added, this list grows and hence the need for the Cisco Configuration Engine to update this list whenever new products are added. This list can be downloaded from the Cisco web site at: http://www.cisco.com.

Step 1 From the Data Manager page, click **Update Product List**.

The Update Product List dialog box appears (see Figure 9-2).

Figure 9-2 Update Product List



Step 2 Select the appropriate download option.

Table 9-2 shows valid values for these fields.

Table 9-2 Valid Values for Update Product List

Attribute	Description	Valid Values
Select Download Option	Available download options	Radio Button
URL	Target URL	Valid URL as per RFC 1738.
Username	Your username	a-z A-Z 0-9 -(hyphen) _ (under-score) . (period)
Password	Your password	Printable characters with a length of 6 – 12

Step 3 Enter the target URL.

Step 4 Enter your username and password.

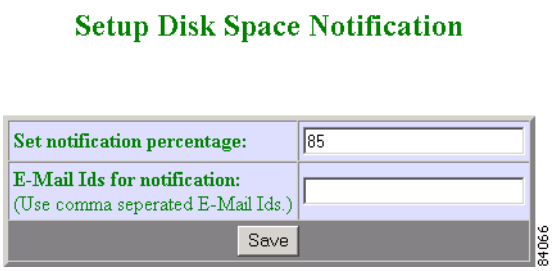
Step 5 To download the product list, click **Download**.

Managing Disk Space

Step 1 From the Data Manager page, click **Manage Disk Space**.

The Setup Disk Space Notification dialog box appears (see Figure 9-3).

Figure 9-3 Disk Space Notification



Step 2 Set the notification percentage to the value that triggers an e-mail notification.

Table 9-3 shows valid values for these fields.

Table 9-3 Valid Values for Setup Disk Space Notification

Attribute	Description	Valid Values
Set notification percentage	Notification percentage that triggers an e-mail notification.	0 – 100
E-Mail Ids for notification:	E-mail address to send notification.	a-z A-Z 0-9 -(hyphen) _ (under-score) . (period)

Step 3 Set the appropriate e-mail address for notification e-mail.

Step 4 To save these entries, click **Save**.

Directory Manager

Note Directory Manager can be accessed only when operating in Internal Directory mode.

To access Directory management tasks, log into the system (see "Logging In" section on page 2-23). Then, from the Home page, click the **Tools** tab. The Tools page appears.

From the Tools page, click **Directory Mgr**.

With the directory manager you can:

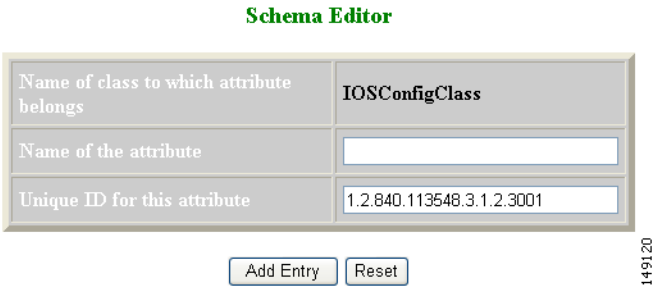
- Edit the schema
- Import a schema from an XML file

Editing Schema

Step 1 From the Directory Manager page, click **Edit Schema**.

The schema editor appears (see Figure 10-1).

Figure 10-1 Schema Editor



Step 2 Enter the name of the new attribute.

Table 10-1 shows valid values for these fields.

Table 10-1 Valid Values for Schema Editor

Attribute	Description	Valid Values
Name of the attribute	Name of the attribute	a-z A-Z 0-9 -(hyphen) _ (under-score) . (period)
Unique ID for this attribute	Unique ID for this attribute	a-z A-Z 0-9 -(hyphen) _ (under-score) . (period)

Step 3 Accept or modify the **Unique ID** for this attribute.

Step 4 To clear your entries, click **Reset**.

Step 5 To add this attribute to the schema, click **Add Entry**.

Importing Schema

You can import a schema accessible from your computer. However, the file must be in XML format and conform to the definitions specified in the document type definition (DTD) file shown here:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- DTD for DAML -->
<!-- Last updated: 2006-01-18 -->
<!ELEMENT attribute EMPTY>
<!ATTLIST attribute
ref CDATA #REQUIRED
required CDATA #REQUIRED
>
<!ELEMENT attribute-type (name, object-identifier, syntax)>
<!ATTLIST attribute-type
id CDATA #REQUIRED
single-value CDATA #REQUIRED
obsolete CDATA #REQUIRED
user-modification CDATA #REQUIRED
>
<!ELEMENT class (name, object-identifier, attribute)>
<!ATTLIST class
id CDATA #REQUIRED
superior CDATA #REQUIRED
type CDATA #REQUIRED
obsolete CDATA #REQUIRED
>
<!ELEMENT directory-schema (attribute-type, class)>
<!ELEMENT dsml (directory-schema)>
```

```
<!ATTLIST dsml
complete CDATA #REQUIRED
>
<!ELEMENT name (#PCDATA)>
<!ELEMENT object-identifier (#PCDATA)>
<!ELEMENT syntax (#PCDATA)>
>
```

Example

For example, a valid schema would look like:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dsml SYSTEM "dsml.dtd">
<dsml complete="true">
    <directory-schema>
        <attribute-type id="IOSe1ipaddress" single-value="true" obsolete="false"
user-modification="true">
            <name>IOSe1ipaddress</name>
            <object-identifier>1.2.840.113548.3.1.2.20</object-identifier>
            <syntax>1.3.6.1.4.1.1466.115.121.1.15</syntax>
        </attribute-type>
        <class id="IOSConfigClass" superior="top" type="structural" obsolete="false">
            <name>IOSConfigClass</name>
            <object-identifier>1.2.840.113548.3.2.2.1</object-identifier>
            <attribute ref="1.2.840.113548.3.1.2.20" required="false"/>
        </class>
    </directory-schema>
</dsml>
```

Step 1 From the Directory Manager page, click **Import Schema**.

The import schema dialog box appears (see Figure 10-2).

Figure 10-2 Import Schema

Step 2 Enter the filename of the schema you want to import in the **Schema Filename** field.

Table 10-2 shows valid values for these fields.

Table 10-2 Valid Values for Import Schema

Attribute	Description	Valid Values
Schema Filename	Name of schema file to import.	a-z A-Z 0-9 -(hyphen) _ (under-score) . (period)

Use the browse function to locate the file, if needed.

Step 3 To clear your entries, click **Reset**.

Step 4 To import the file, click **Import**.

Parameter Manager

To access Parameter management tasks, log into the system (see "Logging In" section on page 2-23). Then, from the Home page, click the **Tools** tab. The Tools page appears.

From the Tools page, click **Parameter Mgr**.

With the directory manager you can:

- Parameter Validations
- Edit Fetch Process
- Edit Save Process
- Import Script File

Parameter Validations

Step 1 From the Parameter Manager page, click **Parameter Validations**.

The Parameter Validations page appears (see Figure 11-1).

Figure 11-1 ParametersValidations Page

Edit Parameters Validations

Available Parameters	Validation Functions
AdminDevType	– No Validation –
IOSdomain	– No Validation –
IOShostname	– No Validation –
IOSipaddress	Positive_integer_Only
IOSpassword	verify_Date_Time
IOSprotocol	verify_Domain_Name
IOSroutingprotocol	verify_Email_Address
IOSsubnetmask	verify_IP_Address
IOStimeout	verify_URL
SW1InterfaceName	– No Validation –
SW2InterfaceName	– No Validation –

Update

129603

Step 2 From drop-down list for each available parameter, select the desired validation function, then click **Update**.

A status page appears showing the updates you have made.

Edit Fetch Process

Step 1 From the Parameter Manager page, click **Edit Fetch Process**.

The Edit Fetch Process page appears (see Figure 11-2).

Figure 11-2 Edit Fetch Process Page

Edit Fetch Process

Fetch Process: – No Fetch Process –

– No Fetch Process –
event_setup.js
event_setup_security.js
fetchP.js
fetchP_no_output.js
saveP.js
saveP_no_output.js

129604

Step 2 Use the drop-down arrow to select the desired fetch process, then click **Update**.

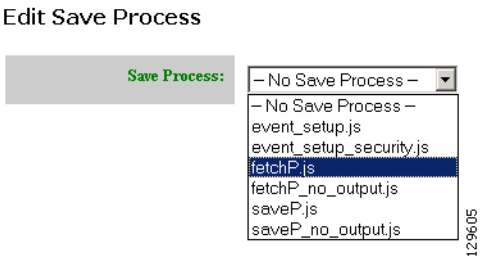
Confirmation of this action is reported.

Edit Save Process

Step 1 From the Parameter Manager page, click **Edit Save Process**.

The Edit Save Process page appears (see Figure 11-3).

Figure 11-3 Edit Save Process Page



Step 2 Use the drop-down arrow to select the desired save process, then click **Update**.

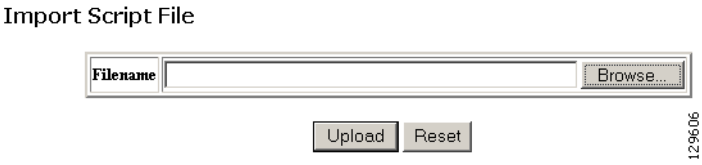
Confirmation of this action is reported.

Import Script File

Step 1 From the Parameter Manager page, click **Import Script File**.

The Import Script File page appears (see Figure 11-4).

Figure 11-4 Import Script File Page



Step 2 Enter the desired filename, or click Browse to access your file system, then click **Upload**.

Templates

When creating a template, it is possible to specify variables that will be contextually substituted. Many of these variables are available in the drop-down menu in the Template Editor (see Figure 12-4). It is also possible to create these files offline without the Template Editor and still use these variables.

The basic format of a template file is simply the text of the configuration to be downloaded to your device (see "Sample Template" section on page 12-125). However, you can put variable substitutions of the following form (for example, the variable name could be *iosipaddress*):

```
Internal directory mode:
   ${LDAP://this:attrName=iosipaddress}
External directory mode:
   ${LDAP://10.1.2.3/cn=Device1,ou=CNSDevices,o=cisco,c=us:attrName=iosipaddress}
```

It is possible to create segments of templates that can be included in other templates. For example, you might have an Ethernet configuration that would be used by multiple devices. In each device template, you could have:

```
#include /opt/CSCOcnsie/Templates/ethernet_setup.cfgtpl
```

Now, you could centralize all the administration for Ethernet configuration in one file.

Caution Circular includes of template files are not allowed.

Sample Template

The following sample is the configuration template for the DemoRouter (*DemoRouter.cfgtpl*), which is pre-loaded on your system:

```
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service udp-small-servers
service tcp-small-servers
!
hostname DemoRouter
!
boot system flash c7200-is-mz
enable secret 5 $1$cMdI$.e37TH540MWB2GW5gMOn3/
enable password cisco
```

```
!
ip subnet-zero
!
interface FastEthernet0/0
 no ip address
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 shutdown
 half-duplex
!
interface Ethernet1/0
 ip address 10.10.1.1 255.255.255.240
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface Ethernet1/1
 no ip address
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 shutdown
!
interface Ethernet1/2
 no ip address
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 shutdown
!
interface Ethernet1/3
 no ip address
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 shutdown
!
ip classless
ip route 0.0.0.0 0.0.0.0 10.10.1.1
ip http server
!
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
!
line con 0
 transport input none
line aux 0
line vty 0 4
 password cisco
 login
!
end
```

Configuration Control Templates

To restart a device with a new image, you need Configuration Control templates that contain the required CLI commands for image activation on particular devices.

For example, if you want to restart a Cisco 3600 Series router with an image named *3600.image*, from the device console, you would issue the following CLI commands:

no boot system
boot system flash:3600.image

The content of the Configuration Control template for image activation should contain the CLI commands that you would normally enter from the device console to activate a new image on the device.

Dynamic Flow Control Template

The inventory information collected from image agents is made available for external users by means of the Dynamic Flow Control Template. This enables you to write templates that can control the flow of configuration and image distribution jobs, based on the inventory information.

Inventory Operations

These are the operations that are exposed to you to access the inventory of the device from the Dynamic Flow Control Templates:

Function	**$!{invObj.getDram()}**
Return Type	int (bytes).
Description	Dram = Main Mem Size + IO Mem Size. Returns the size of the DRAM.

Function	**$!{invObj.getVersionString()}**
Return Type	String.
Description	Returns the version string of the current running image from the device inventory.

Function	**$!{invObj.getImageFile()}**
Return Type	String.
Description	Returns the current running image file name.

Function	**$!{invObj.getImageMD5()}**
Return Type	String.
Description	Returns the MD5 as provided in the device inventory.

Function	**$!{invObj.getStartedAt()}**
Return Type	String.
Description	Returns the time string of when the device started.

Function	**$!{invObj.getPlatformName()}**
Return Type	String.
Description	Returns the platform name.

Function	**$!{invObj.getFlash()}**
Return Type	int (bytes).
Description	Returns the size of the flash.

Function	**$!{invObj.getFileSysSize("bootflash")}**
Return Type	int (bytes).
Description	Returns the size of the bootflash.

Function	**$!{invObj.getFileSysFreespace("bootflash")}**
Return Type	int (bytes).
Description	Returns the amount of free space in the bootflash.

Function	**$!{invObj.getFileSysSize("nvram")}**
Return Type	int (bytes).
Description	Returns the size of the NVRAM.

Function	**$!{invObj.getFileSysFreespace("nvram")}**
Return Type	int (bytes).
Description	Returns the amount of free space in the NVRAM.

Function	**$!{invObj.getFileSysSize("disk0")}**
Return Type	int (bytes).
Description	Returns the size of disk0.

Function	**$!{invObj.getFileSysFreespace("disk0")}**
Return Type	int (bytes).
Description	Returns the amount of free space in disk0.

Function	**$!{invObj.getFileSysSize("slot0")}**
Return Type	int (bytes).
Description	Returns the size of slot0.

Function	**$!{invObj.getFileSysFreespace("slot0")}**
Return Type	int (bytes).
Description	Returns the amount of free space in slot0.

Function	**$!{invObj.getFileSysSize("slot1")}**
Return Type	int (bytes).
Description	Returns the size of slot1.

Function	**$!{invObj.getFileSysFreespace("slot1")}**
Return Type	int (bytes).
Description	Returns the amount of free space in slot1.

Other Operations

These are the operations that are exposed to you to perform an action based on the above criterion from the Dynamic Flow Control Template:

Function	**$!{cnsceObj.distribute()}**
Parameters	None.
Description	Perform image distribution. The pre-configured image is used.

Function	**$!{cnsceObj.activate("persist" | "nv_overwrite")}**
Parameters	Sets the config action: • “persist” – apply and save configuration to NVRAM. • “nv_overwrite” – overwrite NVRAM configuration.
Description	Performs image activation. The pre-configured image is used.

Function	**$!{cnsceObj.updateConfig(true | false, "write" | “persist” | “nv_overwrite”)}**
Parameters	First parameter sets the syntax check: • true – syntax check is turned on. • false – syntax check is turned off. Second parameter is to set the config action: • “write” – apply to running configuration. • “persist” – apply and save configuration to NVRAM. • “nv_overwrite” – overwrite NVRAM configuration.
Description	Performs configuration update. The pre-configured template is used.

Notes

The **invObj.getDram()** operation returns the following:

Dram = Main Mem Size + IO Mem Size

Example

```
#set( $dram = $!{invObj.getDram()} )
##
#if ($dram > 6100)
  $!{cnsceObj.distribute()}
  $!{cnsceObj.activate("persist")}
#end
```

As seen in the example above, you can customize the flow of the job depending on the DRAM size.

When a custom job with the above inventory template is submitted, the device is queried for its inventory, and depending on the DRAM size, the decision is made if the image upgrade is to be performed or not. Hence when the above example inventory template is evaluated, if the DRAM size of the device is greater than 6100 bytes the image distribution and image activation will be performed.

Sample1

```
#set( $dram = $!{invObj.getDram()} )
#set( $flash = $!{invObj.getFlash()} )
##
#if ($dram > 64000000)
    $!{cnsceObj.distribute()}
    #if ( $flash > 48000000 )
       $!{cnsceObj.activate("persist")}
    #end
#end
```

Sample 2

```
#set( $disk0free = $!{invObj.getFileSysFreespace("disk0")} )
##
#if $disk0free > 3500000)
    $!{cnsceObj.distribute()}
     $!{cnsceObj.activate("persist")}
#end
```

Sample 3

```
#set( $flash = $!{invObj.getFlash()} )
##
#if ($flash > 65000000)
    $!{cnsceObj.updateConfig(true, "persist")}
#end
```

Templates for Modular Routers

The template mechanism for the devices has been enhanced to support modular routers. A modular router chassis includes slots in which you can install modules. You can install any module into any available slot in the chassis. Some modules like 2 Ethernet 2 WAN card slot module can in turn have sub slots to install interface cards or line cards. Device management has been extended to support subdevices representing line cards.

Additional attributes representing line card number, line card type, and subdevices have been added to the existing device object structure in the directory server in order to have the same structure to represent the main device or the subdevice.

Currently, card type is a string that maps to the product code of the network module. Since the EPROM data in the card stores part numbers only, not product codes, the part numbers are mapped to product codes. The user uses part numbers and the configuration server maps part numbers to product codes.

In the context of main device, the line card number and line card type fields make no sense and hence are set to NULL value. The subdevices field in the sub device (representing the line card) is set to NULL value.

New interface variable support has been added. These variables are included in the templates, which are parameterize with the interface numbers in the template. These are not attributes. They are special format variables that are replaced by the configuration server based on the interface information, which comes from the device. These variables only specify the relative position of the interface on the module and are replaced by the actual slot number, shelf-ID or port number. The interface variables are wrapped in percent sign (%) characters and specify the type, if any, and the relative position. The configuration server replaces these variables with the interface numbers. The interface type still has to be specified in the CLI using the following syntax:

Interface Variable = %[InterfaceType] RelativePosition%

For example:

%FastEthernet 0% for interface FastEthernet

%Serial 0% interface Serial

%T1 0% controller T1

%E1 0% controller E1

%voice-port 0% voice-port

Example 1:

A network module with two FastEthernet ports plugged in Slot 2 would be referred in the configuration CLI as FastEthernet 2/0 and FastEthernet 2/1 and referred in the template as FastEthernet %FastEthernet 0% and FastEthernet %FastEthernet 1%:

```
!
interface FatsEthernet 2/0
    ip address 10.10.1.1 255.255.255.0
!
interface FatsEthernet 2/1
    ip address 20.20.1.1 255.255.255.0
!
```

Templates for these CLIs would be:

```
!
interface FastEthernet %FastEthernet 0%
    ip address 10.10.1.1 255.255.255.0
!
interface FastEthernet %FastEthernet 1%
    ip address 20.20.1.1 255.255.255.0
!
```

Example 2 (Voice card with two ports plugged in slot 3):

```
!
voice-port 3/0/0
    description 4082224444
!
voice-port 3/0/0
    description 4082225555
!
```

Templates for these CLIs would be:

```
!
voice-port  %voice-port 0%
    description 4082224444
!
voice-port %voice-port 1%
    description 4082225555
!
```

The main device template does not include links to the subdevice templates. The subdevice templates are appended to the main device template. The line card numbers are a parameter in the subdevice templates.

All the CLI commands which reference a line card interface are specified in the subdevice template for that line card. This implies that any command in the global configuration mode, or otherwise, that refers to a particular line card interface is in the template for that subdevice (line card) and not in the main device template.

Only the CLI commands in the global configuration mode, and not pertaining to the any specific interface, are specified in the main device template.

The port number and channel number are not template parameters since these are fixed for a given line card. The network administrator can configure specific channels on the interfaces by explicitly specifying the channels in the subdevice templates.

For example:

interface Serial %Serial 0%:0

Sample Templates for Modular Router

The names of the attributes for slot, slot-unit, line card type and so forth, are used for demonstration purposes.

Main Device Template

```
!
version 12.2
no parser cache
no service single-slot-reload-enable
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname 2600
!
```

```
logging rate-limit console 10 except errors
!
memory-size iomem 25
ip subnet-zero
!
!
!
no ip dhcp-client network-discovery
lcp max-session-starts 0
!
ip classless
no ip http server
!
call rsvp-sync
!
no mgcp timer receive-rtcp
!
mgcp profile default
!
dial-peer cor custom
!
!
!
!
line con 0
line aux 0
line vty 0 4
 login
line vty 5 15
 login
!
```

FastEthernet Template

```
Interface FastEthernet %FastEthernet 0%

ip address 10.0.0.1 255.0.0.0
shutdown
speed auto
```

Voice-port Template

```
voice-port  %voice-port 0%
playout-delay mode adaptive
!
voice-port %voice-port 1%
!
dial-peer voice 10 pots
destination-pattern 200
port %voice-port 0%
forward-digits all


voice-port  %voice-port 0%
!
dial-peer voice 20 pots
destination-pattern 100
port %voice-port 0%
!
voice-port  %voice-port 1%
```

Modular Router Events

Modular router events are published to the event bus and are accessible to applications connected to the bus. The IOS device publishes the system hardware configuration in the *cisco.cns.config.device-details* event after hardware discovery. The Cisco Configuration Engine is configured to listen for this event, retrieve it, and extract the hardware configuration of the device.

Following is the DTD of the *cisco.cns.config.device-details* event that the Cisco IOS device sends:

```
    <!ELEMENT device-details (config-id, connect-interface?, card-info*>
    <!ELEMENT config-id (#PCDATA)>
    <!ELEMENT connect-interface (#PCDATA)>
    <!ELEMENT card-info (card-info+)>
    <!ELEMENT card-info
(card-type,card-desc?,slot,daughter?,serial-number,part-number,hw-version?,board-revision?
,ports?,controller?,rma-number?,test-history?,eeprom-version?,eeprom-data?,interface?,cont
roller?,voice-port?)>
    <!ELEMENT card-type (#PCDATA)>
    <!ELEMENT card-desc (#PCDATA)>
    <!ELEMENT slot (#PCDATA)>
    <!ELEMENT daughter (#PCDATA)>
    <!ELEMENT serial-number (#PCDATA)>
    <!ELEMENT part-number (#PCDATA)>
    <!ELEMENT hw-version (#PCDATA)>
    <!ELEMENT board-revision (#PCDATA)>
    <!ELEMENT ports (#PCDATA)>
    <!ELEMENT controller (#PCDATA)>
    <!ELEMENT rma-number (#PCDATA)>
    <!ELEMENT test-history (#PCDATA)>
    <!ELEMENT eeprom-version (#PCDATA)>
    <!ELEMENT eeprom-data (#PCDATA)>
    <!ELEMENT interface (#PCDATA)>
    <!ELEMENT controller (#PCDATA)>
    <!ELEMENT voice-port (#PCDATA)>
```

Dynamic Templates

There might be times when the actual contents of a template needs to be dynamically generated. To do this, you would use the **#call** mechanism. This executes a JavaScript program whose output becomes part of the template. The program is re-executed each time a device asks for the template.

For example, you might want to distribute the load across the various event gateway processes without permanently assigning a device to a particular event gateway. This is useful because of the limit of 500 devices per event gateway daemon instance.

Let us take the following template as an example:

```
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service udp-small-servers
service tcp-small-servers
!
hostname DemoRouter
#call /opt/CSCOcnsie/Templates/event_setup.js
```

Here is an example of an *event_setup.js* that one might use:

```
/*
 * An instance of Event Gateway resides on every odd port from 11011 to 11031.
```

```
 * This will choose a random one in this range so that devices are spread out
 * evenly among the various ports. Adjust the IP address in the println
 * statement to be the address of the IE2100 itself.
 */
var port = Math.floor(Math.random() * 11) * 2 + 11011;
println("cns event 10.1.6.131 " + port.toString());
```

The result of this combination would be a template that appears as follows:

```
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service udp-small-servers
service tcp-small-servers
!
hostname DemoRouter
cns event 10.1.6.131 11017
```

The last line is programmatically determined and recalculated every time the template is requested by the device. So the next time a device requests this template, the last line might be:

```
cns event 10.1.6.131 11023
```

Simple modifications to *event_setup.js* could even be used to distribute devices across multiple host devices (by dynamically generating the IP address). It could also be used to affect any part of the device configuration—be it DNS servers or routing tables. Anything that is printed out by the JavaScript program becomes a dynamic part of the template.

Control Structures

The configuration template can include simple control structures such as, *if*, *else* and *elseif*. By using these control structures, the user can include or exclude a block of CLI commands based on a parameter stored in the directory.

The syntax for these **#** preprocessing control structures is as follows:

Syntax Description

#if <*URL*> = *constant*

cli-command(s)

#elseif <*URL*> = *constant*

cli-command(s)

#else

cli-command(s)

#endif

Where *constant* is an integer, boolean or a string in single quotes and the <*URL*> is a URL pointing to an attribute in the Directory or Database.

Note Nested **#if** and **#elseif** is NOT supported.

Usage Guidelines

The configuration template can include **#define** entries to define short names for long URLs.

The syntax for the **#define** preprocessing command is as follows:

#define *definition-name* <*URL*> | *constant*

where <*URL*> is a reference to an attribute in the directory.

The configuration template can contain another **#** preprocessing command **#include,** which allows the inclusion of other configuration templates or the results of an ASP page.

The syntax for the **#** preprocessing command is as follows:

#include <*URL*> | '<*Filename*>' | <*Filename*>

Whenever an **#include** directive is encountered, it is replaced by the content of the file.

The following configuration template sample includes either IP sub-template or ISDN sub-template based on the value of the parameter protocol in the directory or database.

Examples

```
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service udp-small-servers
service tcp-small-servers
!
hostname ${LDAP://this:attrName=IOShostname}
#if ${LDAP://this:attrName=IOSIPprotocol} == true then
    #include ${LDAP://this:attrName=IPsubTemplate}
#else
    #include ${LDAP://this:attrName=ISDNsubTemplate}
#endif
```

The parameter, `${LDAP://this:attrName=IPsubTemplate}` contains the location of the file.

Managing Templates

To access Template management tasks, log into the system (see "Logging In" section on page 2-23). Then, from the Home page, click the **Tools** tab. The Tools page appears.

From the Tools page, click **Template Mgr**. The Template Manager page appears showing:

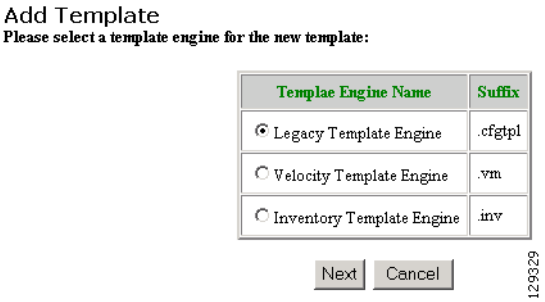
- Add Template
- Edit Template
- Delete Template
- Import Template
- Export Template
- Import Local Template

Adding a Template

Step 1 From the Template Manager page, click **Add Template**.

The Template Engine page appears (see Figure 12-1).

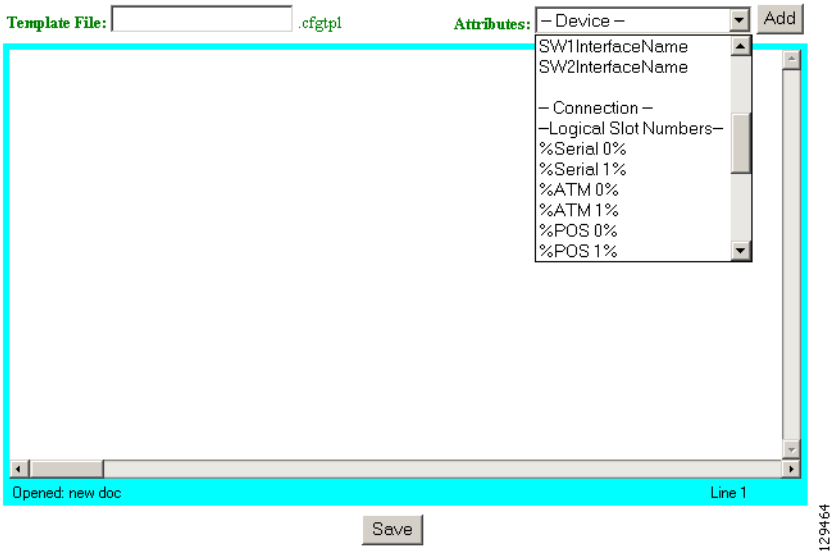
Figure 12-1 Template Engine



Select the Template Engine for the new template, then click **Next**.

A blank template page appears (see Figure 12-2).

Figure 12-2 Blank Template Page



Step 2 Enter the filename for this template in the **Template File** field.

Table 12-1 shows valid values for these fields.

Table 12-1 Valid Values for Add Template

Attribute	Description	Valid Values
Template File	Filename of template	a-z A-Z 0-9 -(hyphen) _ (under-score) . (period)
Attributes	Available attributes	From drop-down list

Step 3 To choose the attributes you want to be included in this template, use the **Attributes** menu.

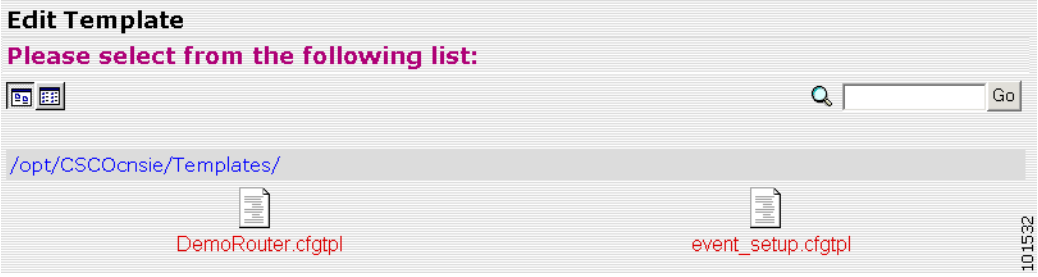
Step 4 To save your entries, click **Save**.

Editing a Template

Step 1 From the Template Manager page, click **Edit Template**.

The Edit Template list appears (see Figure 12-3).

Figure 12-3 Edit Template List



Step 2 Click on the icon for the template file you want to edit.

The template file appears.

Step 3 To edit parameters (attribute information):

a. From the template file page, click **Edit AttributeInfo**.

b. Edit the desired parameter fields.

c. To clear your entries, click **Reset**.

d. To save your changes, click **Save**.

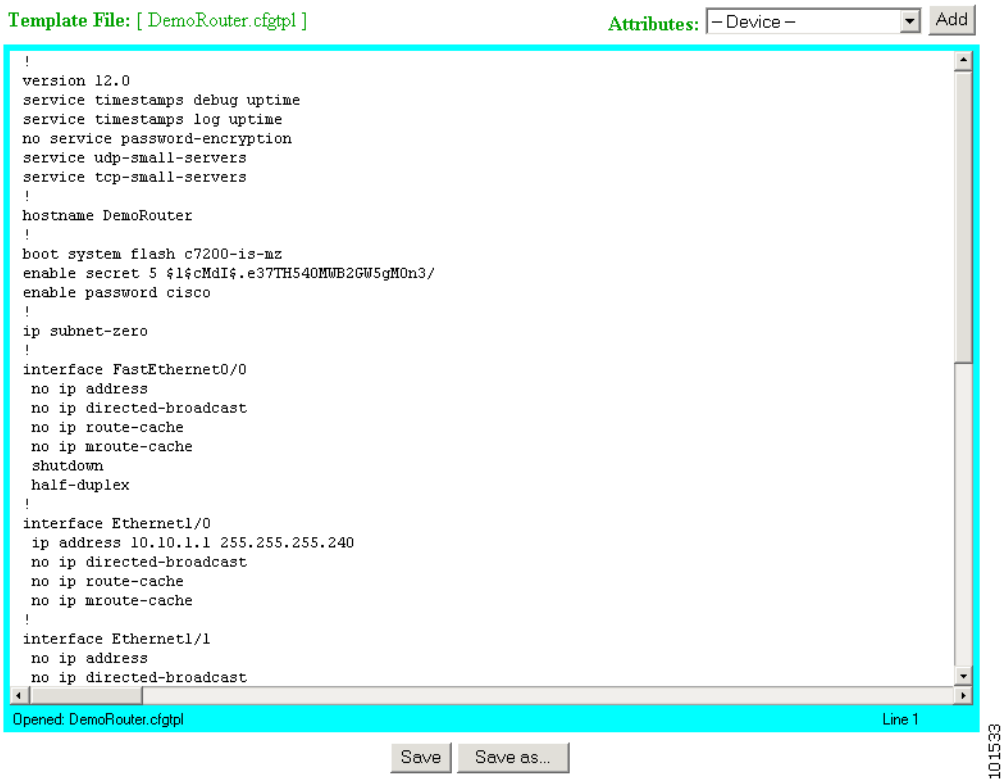
Step 4 To save and apply, **Save and Apply**.

Step 5 To edit template content:

a. To edit the content of a template, from the template file page, click **Edit Content**.

The template content appears (see Figure 12-4).

Figure 12-4 Template Content



b. Edit the content by adding or deleting attributes.

c. To save your edits, click **Save**.

d. To save as a new template, click **Save as**.

Deleting a Template

Step 1 From the Template Manager page, click **Delete Template**.

The template file list appears.

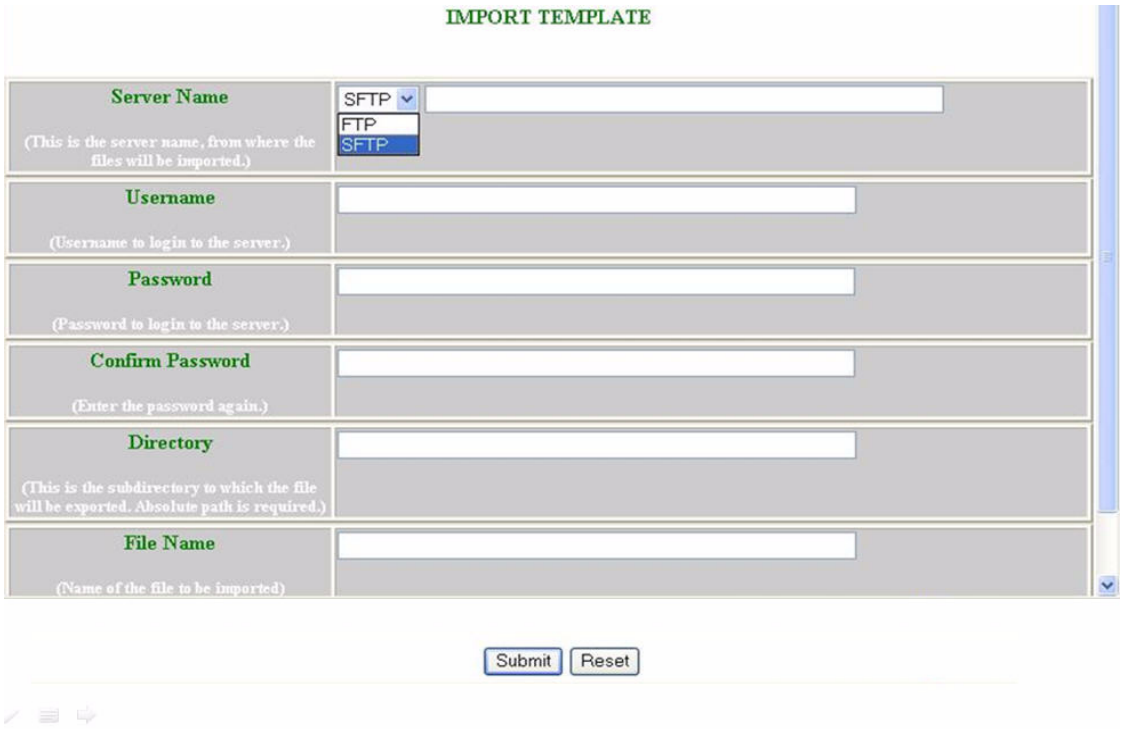
Step 2 Select the template you want to delete.

Step 3 Delete the desired template file.

Importing a Template

Step 1 From the Template Manager page, click **Import Template**.

Figure 12-5 Importing a Template



Step 2 In the dialog box that appears, enter the FTP or SFTP path of the server name in the **FTP/SFTP** field to which the files have to be exported.

Step 3 Enter the username in the **Username** field.

Step 4 Enter the password in the **Password** field.

Step 5 Reenter the password in the **Confirm Password** field**.**

Step 6 Enter the subdirectory path in the **Director** field to which the file should be exported.

Step 7 Enter the name of the template file in the **Filename** field, if known, or browse your directory tree to choose the filename you desire.

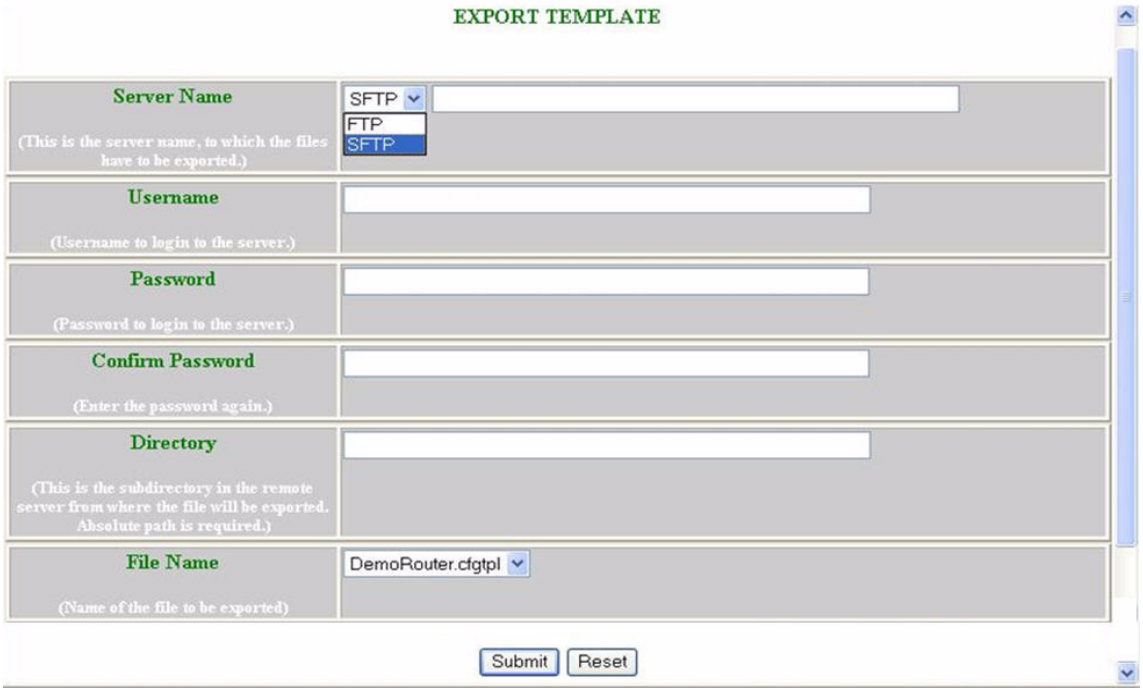
Step 8 To clear the field, click **Reset**.

Step 9 To import the template file, click **Submit**.

Exporting Template

Step 1 From the Template Manager page, click **Export Template**.

Figure 12-6 Exporting a Template



Step 2 In the dialog box that appears, enter the FTP or SFTP path of the server name in the **FTP/SFTP** field to which the files have to be exported.

Step 3 Dnter the username in the **Username** field.

Step 4 Enter the password in the **Password** field.

Step 5 Reenter the password in the **Confirm Password** field**.**

Step 6 Enter the subdirectory path in the **Director** field to which the file should be exported.

Step 7 Enter the name of the template file in the **Filename** field, if known, or browse your directory tree to choose the filename you desire.

Step 8 To clear the field, click **Reset**.

Step 9 To import the femplate file, click **Submit**.

Importing Local Template

Step 1 From the Template Manager page, click **Import Local Template**.

Figure 12-7 Import Local Template

Step 2 Enter the name of the template file in the **Filename** field, if known, or browse your directory tree to choose the filename you prefer.

Step 3 To import the template file, click **Upload**.

Security Manager

With the security manager tool you can change the bootstrap password.

The bootstrap password is used to authenticate a Cisco IOS device before it connects to the Event Gateway. You can set the default bootstarp password by using the Cisco Configuration Engine setup program. For additional information see "Device Authentication" section on page 1-12.

To access Security management tasks, log into the system (see "Logging In" section on page 2-23). Then, from the Home page, click the **Tools** tab. The Tools page appears. From the Tools page, click **Security Mgr**.

The Security Manager page appears showing: BootStrap.

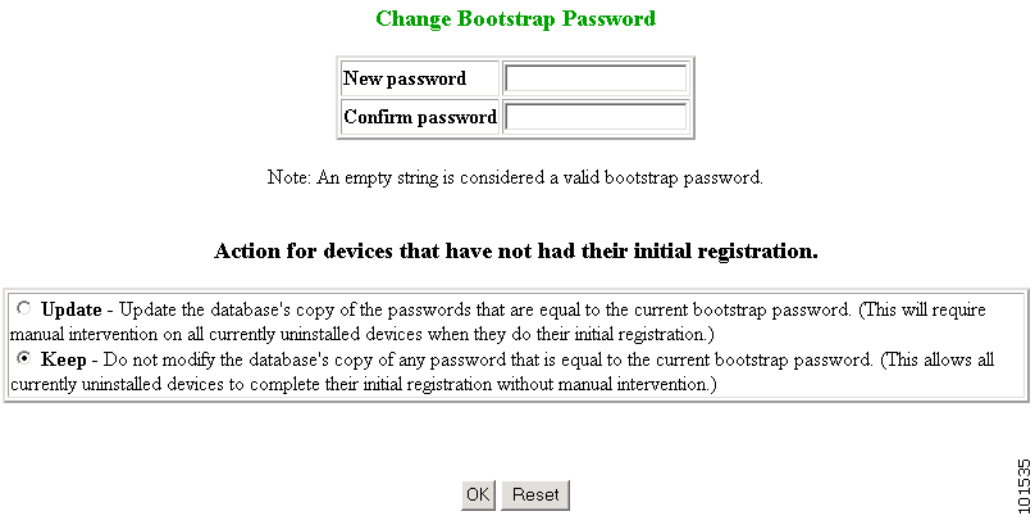
Changing Bootstrap Password

The bootstrap password is used where multiple devices are deployed in a batch. In this case, all devices in a particular batch are given the same (bootstrap) password to use when they each start up on the network for the first time. The bootstrap password can be changed for different batches of devices by using the Security Manager.

Step 1 From the Security Management page, click **BootStrap**.

The Change Bootstrap Password page appears (see Figure 13-1).

Figure 13-1 Change Bootstrap Password



Step 2 In the password dialog box, enter the new password.

Table 13-1 shows valid values for these fields.

Table 13-1 Valid Values for Change Bootstrap Password

Attribute	Description	Valid Values
New password	Bootstrap password	Printable characters with a length of 6 – 12
Confirm password	Bootstrap password	Printable characters with a length of 6 – 12
Update	Modifies the database copy of the password that is equal to the current bootstrap password. This will require manual intervention on all currently uninstalled devices when they do their initial registration.	Radio button
Keep	Does not modify the database copy of any password that is equal to the current bootstrap password. This allows all currently uninstalled devices to complete their initial registration without manual intervention.	Radio button

Step 3 Confirm the new password.

Step 4 Choose (**Keep**, **Update** radio buttons) the subsequent action to the database regarding any password that is equal to the bootstrap password.

Step 5 To clear all entries, click **Reset**.

Step 6 To save the new password, click **OK**.

Log Manager

To access Log management tasks, log into the system (see "Logging In" section on page 2-23). Then, from the Home page, click the **Tools** tab. The Tools page appears.

From the Tools Page, click **Log Manager**. The Log Manager page appears showing:

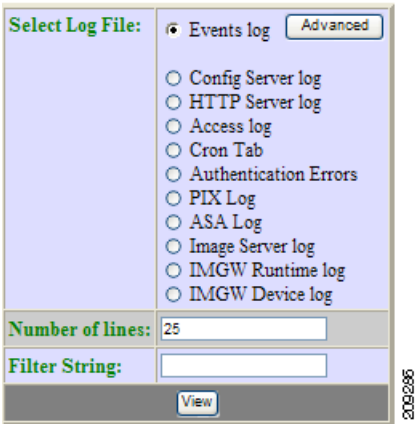
- View Logs
- Clear Logs
- Export Logs
- Change Log Level

Viewing Log Files

Step 1 From the Log Manager page, click **View Logs**.

The View Log Files dialog box appears (see Figure 14-1).

Figure 14-1 Selecting Log File to View



Step 2 Select the log file you want to view.

Table 14-1 shows valid values for these fields.

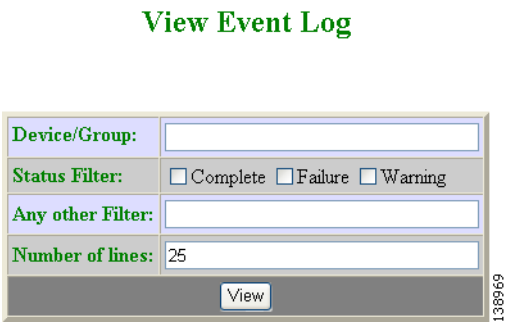
Table 14-1 Valid Values for View Log Files

Attribute	Description	Valid Values
Select Log Files	List of available log files.	Radio button
Number of lines	Number of lines displayed.	
Filter String	Filter string	a-z A-Z 0-9 -(hyphen) _ (under-score) . (period)

Step 3 For additional attributes related to viewing Event Logs, click **Advanced**.

The View Event Log window appears (see Figure 14-2).

Figure 14-2 Event Log Attributes



Step 4 Enter the attributes you want to view a specific Event Log, then click **View**.

Step 5 In the main View Log Files window (see Figure 14-1), set the number lines you want to display.

Step 6 To limit the report to display only specific entries, set a case-sensitive keyword filter, or leave blank.

Step 7 Click **View**.

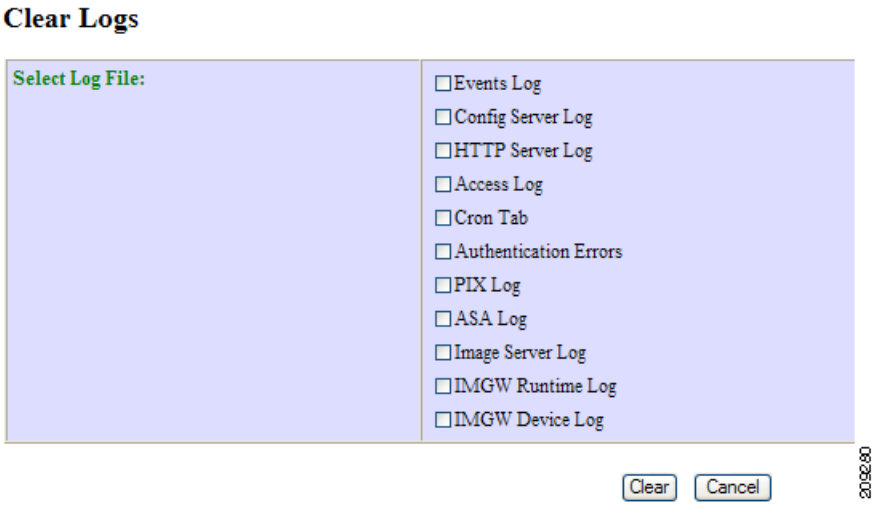
A report displays.

Clearing Logs

Step 1 From the Log Manager page, click **Clear Logs**.

The Clear Log Files dialog box appears (see Figure 14-1).

Figure 14-3 Clear Logs



Step 2 Check the log files you want to clear.

Step 3 To cancel this task, click **Cancel**.

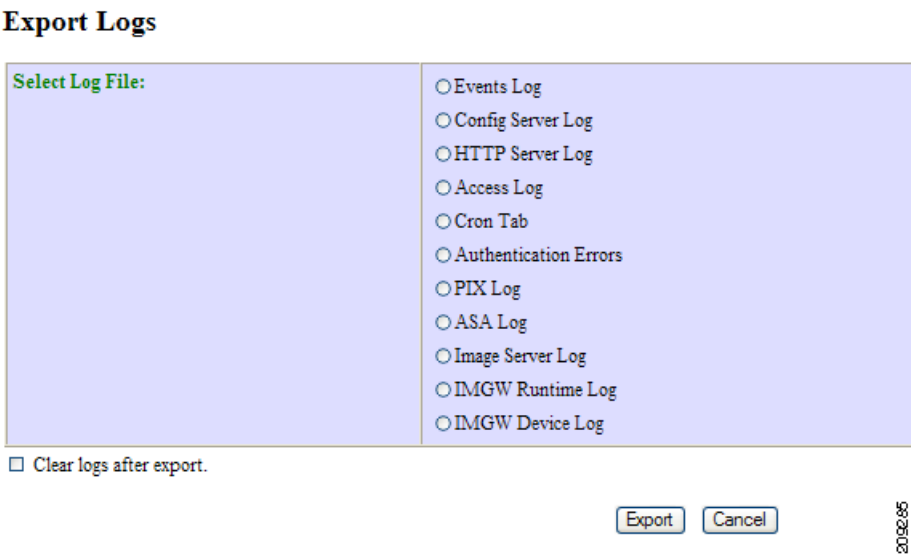
Step 4 To clear the selected log files, click **Clear**.

Exporting Logs

Step 1 From the Log Manager page, click **Export Logs**.

The Export Log Files dialog box appears (see Figure 14-4).

Figure 14-4 Export Logs



Step 2 Check the log files you want to export.

Step 3 To clear logs after export, check the check box.

Step 4 To cancel this task, click **Cancel**.

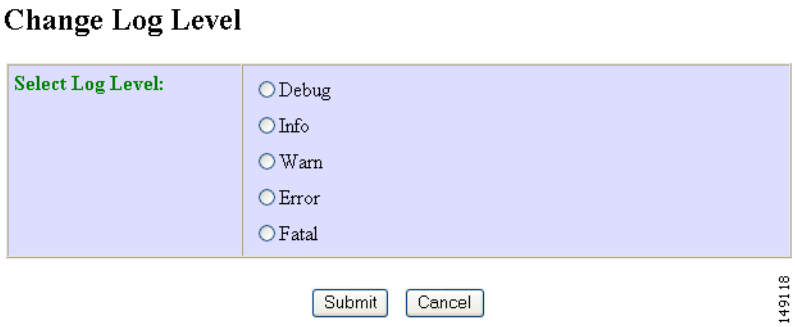
Step 5 To export the selected log files, click **Export**.

Changing Log Level

Step 1 From the Log Manager page, click **Change Log Level**.

The Change Log Level dialog box appears (see Figure 14-5):

Figure 14-5 Selecting Log Level



Step 2 Select the desired log level by clicking the appropriate radio button, then click **Submit**.

Service Manager

To access Service management tasks, log into the system (see "Logging In" section on page 2-23). Then, from the Home page, click the **Tools** tab. The Tools page appears.

From the Tools Page, click **Service Manager**. The Service Manager page appears showing:

- Edit Service Properties
- Edit IMGW Device and Hop Types

Editing Service Properties

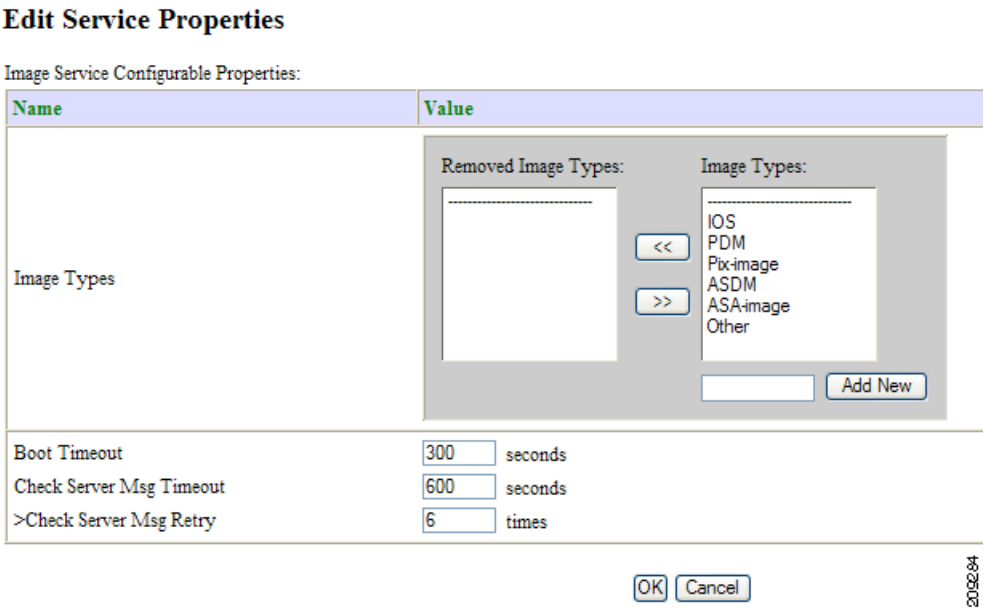
Step 1 From the Service Manager Functional Overview page, click **Edit Service Properties**. The Edit Service Properties page appears (see Figure 15-1).

Figure 15-1 Edit Service Properties

Step 2 From the Edit Service Properties page, select Image Service by clicking the associated radio button.

The service properties page for Image Service appears (see Figure 15-2).

Figure 15-2 Image Service Properties



Step 3 To Edit Image Types: Click the move button (<<) to move an image type to the Removed Image Types column.

Step 4 To Edit Boot Timeout: Enter a new value in the text box.

Step 5 To Edit Check Server Msg Timeout: Enter a new value in the text box.

Step 6 To Edit Check Server Msg Retry: Enter a new value in the text box.

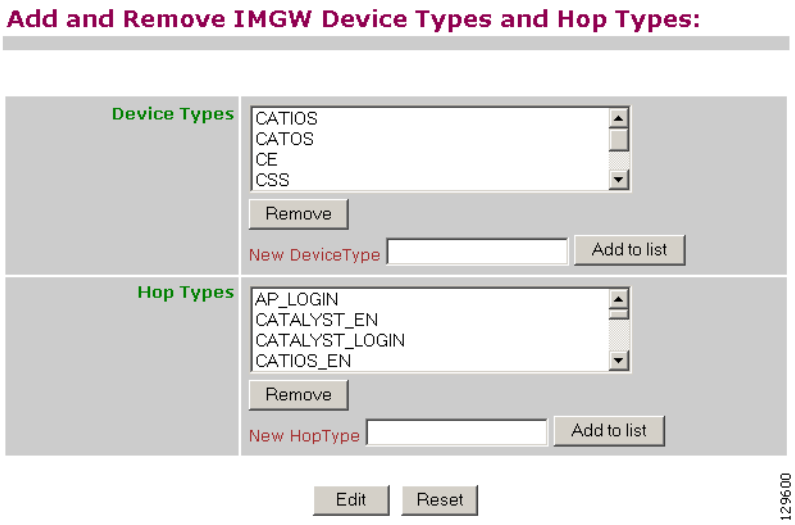
Step 7 To cancel this task, click **Cancel**.

Step 8 To submit the changes, click **OK**.

Editing IMGW Device and Hop Types

Step 1 From the Service Manager Functional Overview page, click **Edit IMGW Device and Hop Types**. The IMGW Device and Hop Types page appears (see Figure 15-3).

Figure 15-3 IMGW Device and Hop Types



Step 2 To remove a Device Type or Hop Type, click the item, then click **Remove**.

Step 3 To add a new Device Type or Hop Type, enter the item in the dialog box, then click **Add to list**.

Step 4 When complete, so save your changes, click **Edit**.

Bulk Data Manager

To access Bulk Data management tasks, log into the system (see "Logging In" section on page 2-23). Then, from the Home page, click the **Tools** tab. The Tools page appears.

From the Tools page, click **Bulk Data Mgr**. The Bulk Data Page appears showing:

- Upload Bulk Data
- Create Sample Data

XML DTD

The following example shows the Document Type Definition (DTD) for the XML bulk upload:

```
<?xml version="1.0" encoding="utf-8"?>
<!--
 * BulkUpdload.dtd - dtd for bulk upload
 *
 * July 2008, Config Engine
 *
 * Copyright (c) 2005-2008, 2011 by cisco Systems, Inc.
 * All rights reserved.
-->

<!ELEMENT cns-bulk-upload (cns-element-data)>
<!ATTLIST cns-bulk-upload
    stop-on-error (true | false) "false"
    version (2.0 | 3.0) "3.0"
>
<!ELEMENT cns-element-data ( NSM-DATA | IMAGE-DATA)>
<!ELEMENT NSM-DATA (cns-device-info*, cns-sub-device-info*, cns-application-info*,
cns-group-info*)>
<!ATTLIST NSM-DATA
    op-type (add|edit|delete) #REQUIRED
    validate-data (true | false) "false"
>
<!ELEMENT cns-device-info (cns-device-name, cns-extended-attr*, dev-image-information?,
imgw-data?)>
<!ELEMENT cns-device-name (#PCDATA)>
<!ATTLIST cns-device-info
        dev-type (other | imgw | pix | asa) "other"
>
<!ELEMENT cns-extended-attr (#PCDATA)>
<!ATTLIST cns-extended-attr
        name CDATA #REQUIRED
>
```

```
<!ELEMENT dev-image-information (image-id, activation-template?, dev-image-info*)>
<!ELEMENT image-id (#PCDATA)>
<!ELEMENT activation-template (#PCDATA)>
<!ELEMENT dev-image-info (image-name, distribution)>
<!ELEMENT image-name (#PCDATA)>
<!ELEMENT distribution ( destination?, location)>
<!ATTLIST distribution
    overwrite  (yes | no) "no"
    erase-flash (yes | no) "no"
    activate (true | false) "false"
>
<!ELEMENT destination (#PCDATA)>
<!ELEMENT location (#PCDATA)>

<!-- Imgw-data-->
<!ELEMENT imgw-data (gateway-id?, device-type?, simulation-agent*, hop-information*)>
<!ELEMENT gateway-id (#PCDATA)>
<!ELEMENT device-type (#PCDATA)>
<!ELEMENT simulation-agent (#PCDATA)>
<!ELEMENT hop-information (hop-type, ip-address?, port?, username?, password?)>
<!ELEMENT hop-type (#PCDATA)>
<!ELEMENT ip-address (#PCDATA)>
<!ELEMENT port (#PCDATA)>
<!ELEMENT username (#PCDATA)>
<!ELEMENT password (#PCDATA)>

<!-- sub-device info-->
<!ELEMENT cns-sub-device-info (cns-sub-device-name, sub-device-id, line-card-type,
cns-extended-attr*, main-device-name?)>
<!ELEMENT cns-sub-device-name (#PCDATA)>
<!ELEMENT sub-device-id (#PCDATA)>
<!ELEMENT line-card-type (#PCDATA)>
<!ELEMENT main-device-name (#PCDATA)>

<!ELEMENT cns-application-info (cns-application-name, cns-subject-mapping*)>
<!ELEMENT cns-application-name (#PCDATA)>
<!ELEMENT cns-subject-mapping (cns-original-subject, cns-pub-mapping*, cns-sub-mapping*,
cns-pub-default, cns-sub-default)>
<!ELEMENT cns-original-subject (#PCDATA)>
<!ELEMENT cns-pub-mapping (#PCDATA)>
<!ELEMENT cns-sub-mapping (#PCDATA)>
<!ELEMENT cns-pub-default (#PCDATA)>
<!ELEMENT cns-sub-default (#PCDATA)>

<!ELEMENT cns-group-info (cns-group-name,cns-group-new-name?, cns-group-member*)>
<!ELEMENT cns-group-name (#PCDATA)>
<!ELEMENT cns-group-new-name (#PCDATA)>
<!ELEMENT cns-group-member (#PCDATA)>
<!ATTLIST cns-group-member
    type  (DEV | GRP) "DEV"
>
<!-- Here starts the definition for Image-data-->
<!ELEMENT IMAGE-DATA (image+)>
<!ATTLIST IMAGE-DATA
    op-type (add|edit|delete) #REQUIRED
    validate-data (true | false) "false"
>
<!ELEMENT image (name, image-info)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT image-info (img-name, img-chksum?, hdr-chksum?, software-version?,
system-description?, file-byte-size?, platform-family-name?, img-location*)>
<!ATTLIST image-info
        image-type (IOS | pix-image | pdm | asa-image | asdm | other) "IOS"
>
```

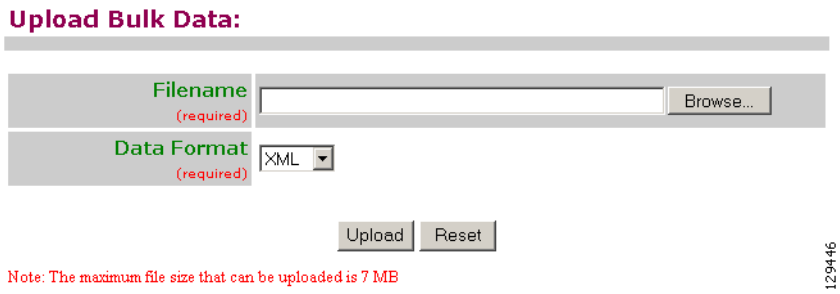
```
<!ELEMENT img-name (#PCDATA)>
<!ELEMENT img-chksum (#PCDATA)>
<!ELEMENT hdr-chksum (#PCDATA)>
<!ELEMENT file-byte-size (#PCDATA)>
<!ELEMENT system-description (#PCDATA)>
<!ELEMENT platform-family-name (#PCDATA)>
<!ELEMENT software-version (#PCDATA)>
<!ELEMENT img-location (#PCDATA)>
```

Uploading Bulk Data

Step 1 From the Bulk Data main menu, click **Upload Bulk Data**.

The Upload Bulk Data parameters page appears (see Figure 16-1).

Figure 16-1 Upload Bulk Data Parameters



Step 2 If you know the filename of the data file you want to load, enter it in the **Filename** field, otherwise use the browse function.

Table 16-1shows the valid values for this field.

Table 16-1 Valid Values for Upload Bulk Data

Attribute	Description	Valid Values
Filename	Name of the file containing the data to be uploaded.	a-z A-Z 0-9 -(hyphen) _ (under-score) . (period)

Step 3 Use the drop-down arrow to select the Data Format:

- XML
- CSV

Step 4 To clear this task, click **Reset**.

Step 5 To upload this data file, click **Upload**.

Command-Line Upload of Bulk Data

You can also upload the XML file to the directory using a command line utility as follows:

FTP the bulk upload XML file to the *$CISCO_CE_INSTALL_ROOT/CSCOdat/scripts* directory on the host system.

Step 6 Log into the box using Telnet.

Step 7 Go to: **$CISCO_CE_INSTALL_ROOT/CSCOdat/scripts**.

Step 8 Run the following command to invoke the bulk upload command line utility:

./upload.sh <*xml filename***>**

For example: **./upload.sh my_bulk_data.xml**

This uploads the data to the LDAP directory.

Using Data Converter Utility

There is a data converter utility that you can use to convert bulk upload data on a system with a release prior to 3.5. This will allow you to do a bulk upload of data to Cisco Configuration Engine.

You can find this utility in <*install base dir*>/ConfigEngine/*CSCOdat/XMLTransform*.

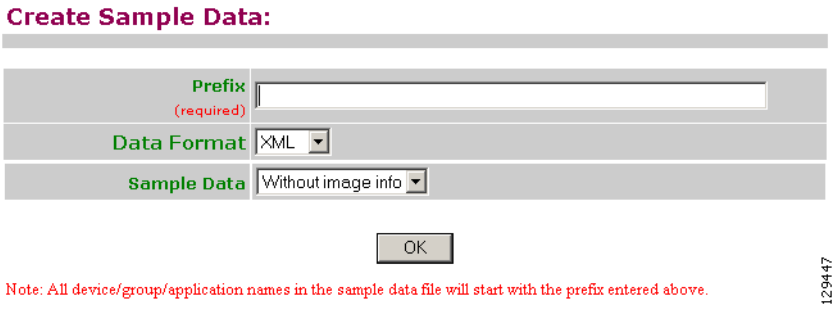
Creating Sample Data

Even though the DTD (see "XML DTD" section on page 16-157) outlines the structure of the input XML file, it does not convey the information about what values should be given for each tag. By looking at the sample data files in this section, you can get an idea of how the data should be arranged in the Bulk Upload XML file.

Step 1 From the Bulk Data main menu, click **Add Bulk Data**.

The Upload Bulk Data page appears (see Figure 16-2).

Figure 16-2 Create Sample Data Page



Step 2 Enter the prefix name for this sample in the **Prefix** field.

Table 16-2 shows valid values for these fields.

Table 16-2 Valid Values for Create Sample Data

Attribute	Description	Valid Values
Prefix	Prefix that is used to create the device/application/group objects.	a-z A-Z 0-9 -(hyphen) _ (under-score) . (period)
Data Format	XML, CSV	From drop-down list
Sample Data Without image info	Creates application, group, device data without the image information for device.	From drop-down list
Sample Data With image info	Creates application, group, device data without the image information for device. Also creates IMAGE object data.	From drop-down list
Sample IMAGE Data only	Creates only IMAGE object data	From drop-down list

Step 3 Select Sample Data.

Step 4 To create this sample, click **OK**.

NSM Data Without Image Info

The following example shows sample device data in XML format:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
  <!DOCTYPE cns-bulk-upload (View Source for full doctype...)>
- <cns-bulk-upload stop-on-error="false" version="3.0">
- <cns-element-data>
- <NSM-DATA op-type="add" validate-data="false">
- <cns-device-info dev-type="other">
  <cns-device-name>myDeviceDevice1</cns-device-name>
  <cns-extended-attr name="IOSconfigtemplate">DemoRouter.cfgtpl</cns-extended-attr>
  <cns-extended-attr name="IOSConfigID">myDeviceDevice1</cns-extended-attr>
  <cns-extended-attr name="IOSEventID">myDeviceDevice1</cns-extended-attr>
- <dev-image-information>
  <image-id>myDeviceDevice1</image-id>
  </dev-image-information>
  </cns-device-info>
- <cns-device-info dev-type="other">
  <cns-device-name>myDeviceDevice2</cns-device-name>
  <cns-extended-attr name="IOSconfigtemplate">DemoRouter.cfgtpl</cns-extended-attr>
  <cns-extended-attr name="parent">/config/myDeviceGroup1</cns-extended-attr>
  <cns-extended-attr name="IOSConfigID">myDeviceDevice2</cns-extended-attr>
  <cns-extended-attr name="IOSEventID">myDeviceDevice2</cns-extended-attr>
- <dev-image-information>
  <image-id>myDeviceDevice2</image-id>
  </dev-image-information>
  </cns-device-info>
- <cns-device-info dev-type="pix">
  <cns-device-name>myDeviceDevice3</cns-device-name>
  <cns-extended-attr name="ErrorAction">revert</cns-extended-attr>
  <cns-extended-attr name="AuthPassword">myDevicepwd</cns-extended-attr>
```

```
  <cns-extended-attr name="IOSconfigtemplate">DemoRouter.cfgtpl</cns-extended-attr>
  <cns-extended-attr name="parent">/config/myDeviceGroup2</cns-extended-attr>
  <cns-extended-attr name="ConfigAction">merge</cns-extended-attr>
  <cns-extended-attr name="IOSConfigID">myDeviceDevice3</cns-extended-attr>
  <cns-extended-attr name="IOSEventID">myDeviceDevice3</cns-extended-attr>
- <dev-image-information>
  <image-id>myDeviceDevice3</image-id>
  </dev-image-information>
  </cns-device-info>
- <cns-device-info dev-type="asa">
  <cns-device-name>myDeviceDevice4</cns-device-name>
  <cns-extended-attr name="ErrorAction">revert</cns-extended-attr>
  <cns-extended-attr name="AuthPassword">myDevicepwd</cns-extended-attr>
  <cns-extended-attr name="IOSconfigtemplate">DemoRouter.cfgtpl</cns-extended-attr>
  <cns-extended-attr
name="parent">/config/myDeviceGroup2/myDeviceSubGroup1</cns-extended-attr>
  <cns-extended-attr name="ConfigAction">merge</cns-extended-attr>
  <cns-extended-attr name="IOSConfigID">myDeviceDevice4</cns-extended-attr>
  <cns-extended-attr name="IOSEventID">myDeviceDevice4</cns-extended-attr>
- <dev-image-information>
  <image-id>myDeviceDevice4</image-id>
  </dev-image-information>
  </cns-device-info>
- <cns-device-info dev-type="imgw">
  <cns-device-name>myDeviceDevice5</cns-device-name>
  <cns-extended-attr name="IOSconfigtemplate">DemoRouter.cfgtpl</cns-extended-attr>
  <cns-extended-attr
name="parent">/config/myDeviceGroup2/myDeviceSubGroup2</cns-extended-attr>
  <cns-extended-attr name="IOSConfigID">myDeviceDevice5</cns-extended-attr>
  <cns-extended-attr name="IOSEventID">myDeviceDevice5</cns-extended-attr>
- <dev-image-information>
  <image-id>myDeviceDevice5</image-id>
  </dev-image-information>
- <imgw-data>
  <gateway-id>myDeviceIMGWGatewayID5</gateway-id>
  <device-type>IOS</device-type>
  <simulation-agent>IMAGEAGENT</simulation-agent>
  <simulation-agent>CONFIGAGENT</simulation-agent>
- <hop-information>
  <hop-type>IOS_LOGIN</hop-type>
  <ip-address>0.0.0.0</ip-address>
  <port>0000</port>
  <username>myDeviceusr5</username>
  <password>myDevicepwd5</password>
  </hop-information>
- <hop-information>
  <hop-type>IOS_EN</hop-type>
  <ip-address />
  <port />
  <username />
  <password>myDevicepasswd5</password>
  </hop-information>
  </imgw-data>
  </cns-device-info>
- <cns-device-info dev-type="imgw">
  <cns-device-name>myDeviceDevice6</cns-device-name>
  <cns-extended-attr name="IOSconfigtemplate">DemoRouter.cfgtpl</cns-extended-attr>
  <cns-extended-attr name="IOSConfigID">myDeviceDevice6</cns-extended-attr>
  <cns-extended-attr name="IOSEventID">myDeviceDevice6</cns-extended-attr>
- <dev-image-information>
  <image-id>myDeviceDevice6</image-id>
  </dev-image-information>
- <imgw-data>
  <gateway-id>myDeviceIMGWGatewayID6</gateway-id>
```

```
  <device-type>IOS</device-type>
  <simulation-agent>IMAGEAGENT</simulation-agent>
  <simulation-agent>CONFIGAGENT</simulation-agent>
- <hop-information>
  <hop-type>IOS_LOGIN</hop-type>
  <ip-address>0.0.0.0</ip-address>
  <port>0000</port>
  <username>myDeviceusr6</username>
  <password>myDevicepwd6</password>
  </hop-information>
- <hop-information>
  <hop-type>IOS_EN</hop-type>
  <ip-address />
  <port />
  <username />
  <password>myDevicepasswd6</password>
  </hop-information>
  </imgw-data>
  </cns-device-info>
- <cns-group-info>
  <cns-group-name>/config/myDeviceGroup1</cns-group-name>
  <cns-group-member type="GRP">myDeviceSubGroup1</cns-group-member>
  </cns-group-info>
- <cns-group-info>
  <cns-group-name>/config/myDeviceGroup2</cns-group-name>
  <cns-group-member type="GRP">myDeviceSubGroup1</cns-group-member>
  <cns-group-member type="GRP">myDeviceSubGroup2</cns-group-member>
  </cns-group-info>
  </NSM-DATA>
  </cns-element-data>
  </cns-bulk-upload>
```


NSM Data Sample With Image Info

The following example shows sample data with the image in XML format:


```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
  <!DOCTYPE cns-bulk-upload (View Source for full doctype...)>
- <cns-bulk-upload stop-on-error="false" version="3.0">
- <cns-element-data>
- <NSM-DATA op-type="add" validate-data="false">
- <cns-device-info dev-type="other">
  <cns-device-name>myDeviceDevice1</cns-device-name>
  <cns-extended-attr name="IOSconfigtemplate">DemoRouter.cfgtpl</cns-extended-attr>
  <cns-extended-attr name="IOSConfigID">myDeviceDevice1</cns-extended-attr>
  <cns-extended-attr name="IOSEventID">myDeviceDevice1</cns-extended-attr>
- <dev-image-information>
  <image-id>myDeviceDevice1</image-id>
  <activation-template>DemoRouter.cfgtpl</activation-template>
- <dev-image-info>
  <image-name>myDeviceIMAGEObj1</image-name>
- <distribution activate="false" erase-flash="no" overwrite="yes">
  <destination>flash</destination>
  <location>tftp://test.com/c7200-js-mz1</location>
  </distribution>
  </dev-image-info>
  </dev-image-information>
  </cns-device-info>
- <cns-device-info dev-type="other">
  <cns-device-name>myDeviceDevice2</cns-device-name>
```

```
  <cns-extended-attr name="IOSconfigtemplate">DemoRouter.cfgtpl</cns-extended-attr>
  <cns-extended-attr name="parent">/config/myDeviceGroup1</cns-extended-attr>
  <cns-extended-attr name="IOSConfigID">myDeviceDevice2</cns-extended-attr>
  <cns-extended-attr name="IOSEventID">myDeviceDevice2</cns-extended-attr>
- <dev-image-information>
  <image-id>myDeviceDevice2</image-id>
  <activation-template>DemoRouter.cfgtpl</activation-template>
- <dev-image-info>
  <image-name>myDeviceIMAGEObj2</image-name>
- <distribution activate="false" erase-flash="no" overwrite="yes">
  <destination>flash</destination>
  <location>tftp://test.com/c7200-js-mz2</location>
  </distribution>
  </dev-image-info>
  </dev-image-information>
  </cns-device-info>
- <cns-device-info dev-type="pix">
  <cns-device-name>myDeviceDevice3</cns-device-name>
  <cns-extended-attr name="ErrorAction">revert</cns-extended-attr>
  <cns-extended-attr name="AuthPassword">myDevicepwd</cns-extended-attr>
  <cns-extended-attr name="IOSconfigtemplate">DemoRouter.cfgtpl</cns-extended-attr>
  <cns-extended-attr name="parent">/config/myDeviceGroup2</cns-extended-attr>
  <cns-extended-attr name="ConfigAction">merge</cns-extended-attr>
  <cns-extended-attr name="IOSConfigID">myDeviceDevice3</cns-extended-attr>
  <cns-extended-attr name="IOSEventID">myDeviceDevice3</cns-extended-attr>
- <dev-image-information>
  <image-id>myDeviceDevice3</image-id>
  <activation-template>DemoRouter.cfgtpl</activation-template>
- <dev-image-info>
  <image-name>myDeviceIMAGEObj3</image-name>
- <distribution activate="false" erase-flash="no" overwrite="yes">
  <destination>flash</destination>
  <location>tftp://test.com/c7200-js-mz3</location>
  </distribution>
  </dev-image-info>
  </dev-image-information>
  </cns-device-info>
- <cns-device-info dev-type="asa">
  <cns-device-name>myDeviceDevice4</cns-device-name>
  <cns-extended-attr name="ErrorAction">revert</cns-extended-attr>
  <cns-extended-attr name="AuthPassword">myDevicepwd</cns-extended-attr>
  <cns-extended-attr name="IOSconfigtemplate">DemoRouter.cfgtpl</cns-extended-attr>
  <cns-extended-attr
name="parent">/config/myDeviceGroup2/myDeviceSubGroup1</cns-extended-attr>
  <cns-extended-attr name="ConfigAction">merge</cns-extended-attr>
  <cns-extended-attr name="IOSConfigID">myDeviceDevice4</cns-extended-attr>
  <cns-extended-attr name="IOSEventID">myDeviceDevice4</cns-extended-attr>
- <dev-image-information>
  <image-id>myDeviceDevice4</image-id>
  <activation-template>DemoRouter.cfgtpl</activation-template>
- <dev-image-info>
  <image-name>myDeviceIMAGEObj4</image-name>
- <distribution activate="false" erase-flash="no" overwrite="yes">
  <destination>flash</destination>
  <location>tftp://test.com/c7200-js-mz4</location>
  </distribution>
  </dev-image-info>
  </dev-image-information>
  </cns-device-info>
- <cns-device-info dev-type="imgw">
  <cns-device-name>myDeviceDevice5</cns-device-name>
  <cns-extended-attr name="IOSconfigtemplate">DemoRouter.cfgtpl</cns-extended-attr>
  <cns-extended-attr
name="parent">/config/myDeviceGroup2/myDeviceSubGroup2</cns-extended-attr>
```

```
  <cns-extended-attr name="IOSConfigID">myDeviceDevice5</cns-extended-attr>
  <cns-extended-attr name="IOSEventID">myDeviceDevice5</cns-extended-attr>
- <dev-image-information>
  <image-id>myDeviceDevice5</image-id>
  <activation-template>DemoRouter.cfgtpl</activation-template>
- <dev-image-info>
  <image-name>myDeviceIMAGEObj5</image-name>
- <distribution activate="false" erase-flash="no" overwrite="yes">
  <destination>flash</destination>
  <location>tftp://test.com/c7200-js-mz5</location>
  </distribution>
  </dev-image-info>
  </dev-image-information>
- <imgw-data>
  <gateway-id>myDeviceIMGWGatewayID5</gateway-id>
  <device-type>IOS</device-type>
  <simulation-agent>IMAGEAGENT</simulation-agent>
  <simulation-agent>CONFIGAGENT</simulation-agent>
- <hop-information>
  <hop-type>IOS_LOGIN</hop-type>
  <ip-address>0.0.0.0</ip-address>
  <port>0000</port>
  <username>myDeviceusr5</username>
  <password>myDevicepwd5</password>
  </hop-information>
- <hop-information>
  <hop-type>IOS_EN</hop-type>
  <ip-address />
  <port />
  <username />
  <password>myDevicepasswd5</password>
  </hop-information>
  </imgw-data>
  </cns-device-info>
- <cns-device-info dev-type="imgw">
  <cns-device-name>myDeviceDevice6</cns-device-name>
  <cns-extended-attr name="IOSconfigtemplate">DemoRouter.cfgtpl</cns-extended-attr>
  <cns-extended-attr name="IOSConfigID">myDeviceDevice6</cns-extended-attr>
  <cns-extended-attr name="IOSEventID">myDeviceDevice6</cns-extended-attr>
- <dev-image-information>
  <image-id>myDeviceDevice6</image-id>
  <activation-template>DemoRouter.cfgtpl</activation-template>
- <dev-image-info>
  <image-name>myDeviceIMAGEObj6</image-name>
- <distribution activate="false" erase-flash="no" overwrite="yes">
  <destination>flash</destination>
  <location>tftp://test.com/c7200-js-mz6</location>
  </distribution>
  </dev-image-info>
  </dev-image-information>
- <imgw-data>
  <gateway-id>myDeviceIMGWGatewayID6</gateway-id>
  <device-type>IOS</device-type>
  <simulation-agent>IMAGEAGENT</simulation-agent>
  <simulation-agent>CONFIGAGENT</simulation-agent>
- <hop-information>
  <hop-type>IOS_LOGIN</hop-type>
  <ip-address>0.0.0.0</ip-address>
  <port>0000</port>
  <username>myDeviceusr6</username>
  <password>myDevicepwd6</password>
  </hop-information>
- <hop-information>
  <hop-type>IOS_EN</hop-type>
```

```
  <ip-address />
  <port />
  <username />
  <password>myDevicepasswd6</password>
  </hop-information>
  </imgw-data>
  </cns-device-info>
- <cns-group-info>
  <cns-group-name>/config/myDeviceGroup1</cns-group-name>
  <cns-group-member type="GRP">myDeviceSubGroup1</cns-group-member>
  </cns-group-info>
- <cns-group-info>
  <cns-group-name>/config/myDeviceGroup2</cns-group-name>
  <cns-group-member type="GRP">myDeviceSubGroup1</cns-group-member>
  <cns-group-member type="GRP">myDeviceSubGroup2</cns-group-member>
  </cns-group-info>
  </NSM-DATA>
  </cns-element-data>
  </cns-bulk-upload>
```

NOTES

- For Bulk Upload of NSM devices with Image Info, make sure that the image objects referenced in the **dev-image-info** element tag already exist.
- The location given should be one of the multiple image locations specified with the image object.
- If there are errors while adding the devices, please check the error file provided as a result of the Upload operation. There can be an exception given as CISException, which points to the CISDevice creation failed, which could have occurred if you had ignored the checklist. In this case, just recheck the information provided in the **dev-image-information** element tag. Correct the file and upload it again.

Image Sample Data

The following example shows image data sample:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
  <!DOCTYPE cns-bulk-upload (View Source for full doctype...)>
- <cns-bulk-upload stop-on-error="false" version="3.0">
- <cns-element-data>
- <IMAGE-DATA op-type="add" validate-data="false">
- <image>
  <name>myDeviceIMAGEObj1</name>
- <image-info image-type="IOS">
  <img-name>c7200-js-mz1</img-name>
  <img-chksum>0x1256faf245</img-chksum>
  <software-version>12.2(8)T6</software-version>
  <system-description>Cisco Network Operating System</system-description>
  <file-byte-size>1040</file-byte-size>
  <platform-family-name>7200</platform-family-name>
  <img-location>tftp://test.com/c7200-js-mz1</img-location>
  </image-info>
  </image>
- <image>
  <name>myDeviceIMAGEObj2</name>
- <image-info image-type="IOS">
  <img-name>c7200-js-mz2</img-name>
  <img-chksum>0x1256faf245</img-chksum>
  <software-version>12.2(8)T6</software-version>
```

```
  <system-description>Cisco Network Operating System</system-description>
  <file-byte-size>1040</file-byte-size>
  <platform-family-name>7200</platform-family-name>
  <img-location>tftp://test.com/c7200-js-mz2</img-location>
  </image-info>
  </image>
- <image>
  <name>myDeviceIMAGEObj3</name>
- <image-info image-type="pix-image">
  <img-name>c7200-js-mz3</img-name>
  <img-chksum>0x1256faf245</img-chksum>
  <software-version>12.2(8)T6</software-version>
  <system-description>Cisco Network Operating System</system-description>
  <file-byte-size>1040</file-byte-size>
  <platform-family-name>7200</platform-family-name>
  <img-location>tftp://test.com/c7200-js-mz3</img-location>
  </image-info>
  </image>
- <image>
  <name>myDeviceIMAGEObj4</name>
- <image-info image-type="pdm">
  <img-name>c7200-js-mz4</img-name>
  <img-chksum>0x1256faf245</img-chksum>
  <software-version>12.2(8)T6</software-version>
  <system-description>Cisco Network Operating System</system-description>
  <file-byte-size>1040</file-byte-size>
  <platform-family-name>7200</platform-family-name>
  <img-location>tftp://test.com/c7200-js-mz4</img-location>
  </image-info>
  </image>
- <image>
  <name>myDeviceIMAGEObj5</name>
- <image-info image-type="asa-image">
  <img-name>c7200-js-mz5</img-name>
  <img-chksum>0x1256faf245</img-chksum>
  <software-version>12.2(8)T6</software-version>
  <system-description>Cisco Network Operating System</system-description>
  <file-byte-size>1040</file-byte-size>
  <platform-family-name>7200</platform-family-name>
  <img-location>tftp://test.com/c7200-js-mz5</img-location>
  </image-info>
  </image>
- <image>
  <name>myDeviceIMAGEObj6</name>
- <image-info image-type="asdm">
  <img-name>c7200-js-mz6</img-name>
  <img-chksum>0x1256faf245</img-chksum>
  <software-version>12.2(8)T6</software-version>
  <system-description>Cisco Network Operating System</system-description>
  <file-byte-size>1040</file-byte-size>
  <platform-family-name>7200</platform-family-name>
  <img-location>tftp://test.com/c7200-js-mz6</img-location>
  </image-info>
  </image>
- <image>
  <name>myDeviceIMAGEObj7</name>
- <image-info image-type="IOS">
  <img-name>c7200-js-mz7</img-name>
  <img-chksum>0x1256faf245</img-chksum>
  <software-version>12.2(8)T6</software-version>
  <system-description>Cisco Network Operating System</system-description>
  <file-byte-size>1040</file-byte-size>
  <platform-family-name>7200</platform-family-name>
  <img-location>tftp://test.com/c7200-js-mz7</img-location>
```

```
  </image-info>
  </image>
- <image>
  <name>myDeviceIMAGEObj8</name>
- <image-info image-type="IOS">
  <img-name>c7200-js-mz8</img-name>
  <img-chksum>0x1256faf245</img-chksum>
  <software-version>12.2(8)T6</software-version>
  <system-description>Cisco Network Operating System</system-description>
  <file-byte-size>1040</file-byte-size>
  <platform-family-name>7200</platform-family-name>
  <img-location>tftp://test.com/c7200-js-mz8</img-location>
  </image-info>
  </image>
  </IMAGE-DATA>
  </cns-element-data>
  </cns-bulk-upload>
```

Email Manager

To access Email management tasks, log into the system (see "Logging In" section on page 2-23). Then, from the Home page, click the **Tools** tab. The Tools page appears.

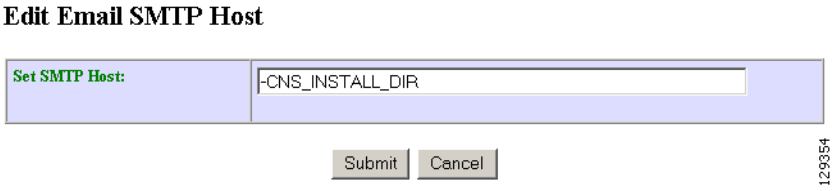
From the Tools page, click **Email Manager**. The Email page appears showing: Edit Email SMTP Host.

Editing Email SMTP Host

Step 1 From the Email Manager Functional Overview page, click **Edit Email SMTP Host**.

The Edit Email SMTP Host page appears:

Figure 17-1 Edit Email SMTP Host



Step 2 Enter a new host path, then click **Submit**.

Image Service

This chapter describes Image Service management tasks. To access the Image Service feature, click the **Image Service** tab. The Image Service Functional Overview page appears showing:

- Images
- Search Parameters

Note For External Directory Mode, the Search Parameters tab is called Preconditions.

Working with Images

From the Image Service Functional Overview page, click **Images**. The Images Functional Overview page appears showing:

- View Image
- Create Image
- Edit Image
- Delete Image
- Associate Image with Device(s)

Viewing an Image

Step 1 From the Images Functional Overview page, click **View Image**.

The list of images to view appears (see Figure 18-1).

Figure 18-1 View Image List

View Image

Search : Go

Name	Image Locations
image1	ftp://ftp:test@10.1.7.24/tftp/c7200-is-mz.123-1.9.T
image2	ftp://ftp:test@10.1.7.24/tftp/c3640-tea-mz.geo_20030810
image3	ftp://ftp:test@10.1.7.24/tftp/c7200-tk8ea-mz.geo_20030721.T
image4	ftp://ftp:test@10.1.7.24/tftp/c7200-tk8ea-mz.v123-3_20030714.T

101545

Step 2 From the Name column, select the image you want to view.

The image information appears (see Figure 18-2).

Figure 18-2 View Image Information

View Image

image1	
Image Name	C7200-IS-MZ
Version	12.3(1.9)T,
Platform Family	C7200
Image Checksum	8fc6160c10141ed4122b6db19f01d2f0
Size	17723372 bytes
Description	Cisco Internetwork Operating System Software IOS (tm) 7200 Software (C7200-IS-MZ), Version 12.3(1.9)T, MAINTENANCE INTERIM SOFTWARE Synched to technology version 12.3(1.9) TAC Support: http://www.cisco.com/tac Copyright (c) 1986-2003 by cisco Systems, Inc. Compiled Thu 12-Jun-03 17:19 by ccai
Image Type	IOS
Image Locations	ftp://ftp:test@10.1.7.24/tftp/c7200-is-mz.123-1.9.T

101546

Adding an Image

Step 1 From the Image Service Functional Overview page, click **Create Image**.

The Create Image page appears (see Figure 18-3).

Figure 18-3 Create Image

Create Image

Name (required)	
Image Name	
Version	
Platform Family	
Image Checksum	
Size (required)	
Description	
Image Type	IOS
Image Locations	
	Add Another Row
	Enter a location as <protocol>://<hostname><absolutefilepath> For example: ftp://username:password@ftp.server.com/directory/imagefile

Populate image attributes by acquiring values from image location Populate

Lookup image attributes from CCO

Create Cancel

101547

There are two methods for creating an Image Object:

Manual data entry

To enter image information manually, jump to Step 2.

Timesaver You can get image attributes for manual entry by clicking the link: **Lookup image attributes from Cisco.com**.

Automatic data entry

a. In the **Image Location** field, enter a valid URL for the desired image.

b. Click **Populate**.

Step 2 Enter the name of the image used by Image Service to identify this image object in the **Name** field.

Table 18-1 shows valid values for these attributes.

Table 18-1 Valid Values for Create Image

Attribute	Description	Valid Values
Name	The name used my Image Services to identify this image object.	a-z A-Z 0-9 # _ (under-score) - (hyphen)
Image Name	The actual Image name.	a-z A-Z 0-9 - (hyphen)
Version	Version of the image.	a-z A-Z 0-9 . (period) ((open braces)) (close braces)
Platform Family	Platform family of the image.	a-z A-Z 0-9 - (hyphen)
Image Checksum	Checksum generated by MD5 hashing algorithm	128-bit hex number
Size	File size	0 – 9
Description	Description of the image.	Any text except Ctrl characters.
Image Type	(i) PDM (ii) QDM (iii) VDM (iv) Other (v) Pix-image	From drop-down list.
Image Location	- Any Valid URL: (i) http (ii) https (iii) ftp (iv) tftp - rcp	Valid URL as per RFC 1738.

Step 3 Enter the actual image name in the **Image Name** field.

Step 4 Enter the version of the image in the **Version** field.

Step 5 Enter the name of the platform family in the **Platform Family** field.

Step 6 Enter the image checksum for the image in the **Image Checksum** field.

Step 7 Enter the size of this file in the **Size** field.

Step 8 Enter a description of the image in the space provided.

Step 9 Select an image type from the **Image Type** drop-down list.

Step 10 Enter a valid URL for the image location in the **Image Location** field.

Follow the proper syntax as described.

Note You can create an image without specifying a location. You can add a location later by using the **Edit Image** function.

Step 11 To add another row for image location, click **Add Another Row**.

You can locate multiple copies of an image on separate servers. This allows you to do load-sharing when updating a large number of devices. Each device in a large group can be associated with a copy of the image (see"Adding Devices" section on page 3-31) located at one of many server locations.

Step 12 To cancel this task, click **Cancel**.

Step 13 To create this image, click **Create**.

Editing an Image

Step 1 From the Image Service Functional Overview page, click **Edit Image**.

The Edit Image page appears (see Figure 18-4).

Figure 18-4 Edit Image

Edit Image

Search : Go

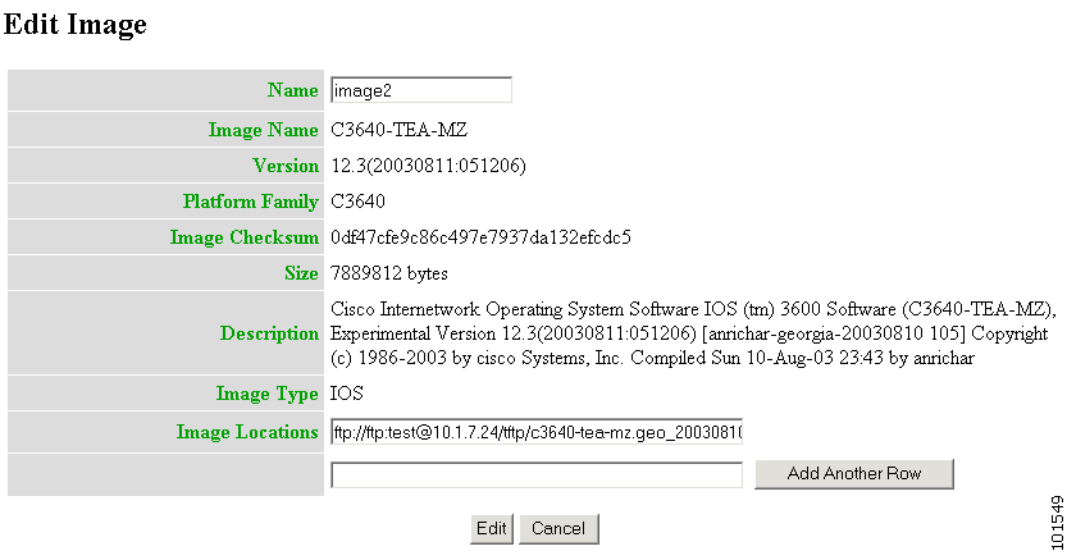
Name	Image Locations
image1	ftp://ftp:test@10.1.7.24/tftp/c7200-is-mz.123-1.9.T
image2	ftp://ftp:test@10.1.7.24/tftp/c3640-tea-mz.geo_20030810
image3	ftp://ftp:test@10.1.7.24/tftp/c7200-tk8ea-mz.geo_20030721.T
image4	ftp://ftp:test@10.1.7.24/tftp/c7200-tk8ea-mz.v123-3_20030714.T

101548

Step 2 Select the image you want to edit by clicking the Image Name.

The Edit Image information page appears (see).

Figure 18-5 Edit Image Information



Step 3 To edit the image name, enter a new value in the **Name** field.

Table 18-2 Valid Values for Edit Image

Attribute	Description	Valid Values
Name	The name used my Image Services to identify this image object.	a-z A-Z 0-9 # _ (under-score) - (hyphen)
Image Location	- Any Valid URL: (i) http (ii) https (iii) ftp (iv) tftp - rcp	Valid URL as per RFC 1738.

Step 4 To edit the image location, enter a valid URL in the **Image Location** field.

Step 5 To cancel this task, click **Cancel**.

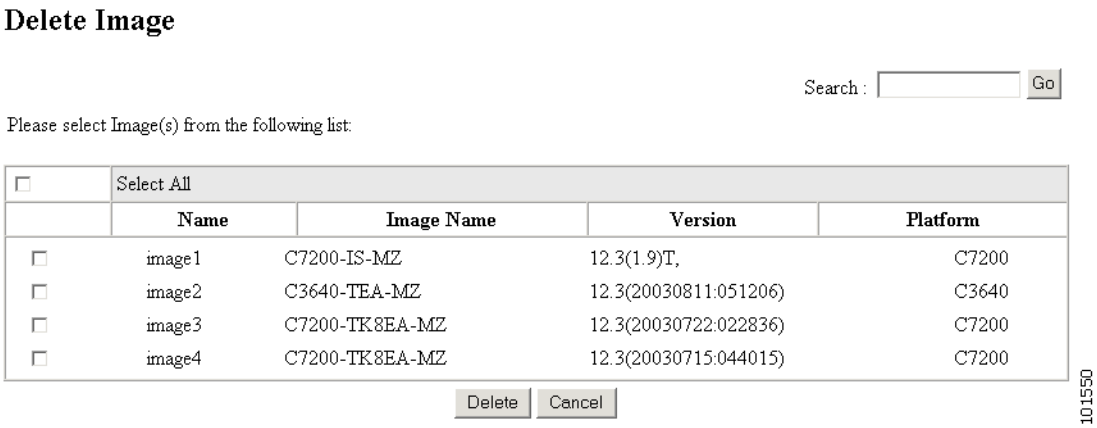
Step 6 To make these changes, click **Edit**.

Deleting an Image

Step 1 From the Image Service Functional Overview page, click **Delete Image**.

The Delete Image page appears (see Figure 18-6).

Figure 18-6 Delete Image



Step 2 Check the image(s) you want to delete.

Step 3 To cancel this task, click **Cancel**.

Step 4 To make these changes, click **Delete**.

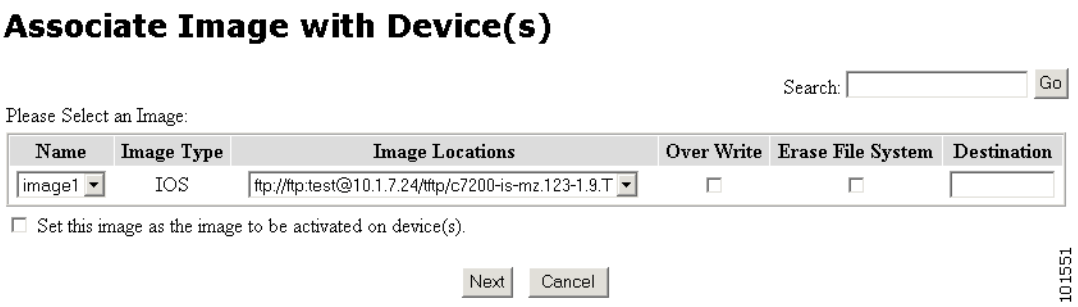
Associating Images with Devices

Note To associate a device with the image, the device must have been registered for image service during device object creation by providing an ImageID. If this has not been done, before trying to associate the device, the device must be edited and an ImageID must be provided.

Step 1 From the Image Service Functional Overview page, click **Associate Image with Device(s)**.

The Associate Image with Device(s) page appears (see Figure 18-7).

Figure 18-7 Associate Image with Device(s)



Step 2 Select the image from the **Name** drop-down list.

The **Image Type** field and **Image Location** drop-down box are populated with corresponding information for the image.

Step 3 From the **Image Location** drop-down list, select the desired location.

Step 4 In the Destination field, enter a valid URL where the image will be copied.

For example:

disk0:/c7200-mz

Step 5 To assign this image to be the active image after distribution, check **Set this image as the Image to be activated on device**.

Step 6 To cancel this task, click **Cancel**.

Step 7 To continue, click **Next**.

The Group list page appears.

Step 8 To associate this image with a group of devices, check the group, then click **Submit**.

Step 9 To associate this image with specific devices, click **View**.

The Device list page appears (see Figure 18-8).

Figure 18-8 Device List

Step 10 Check the desired device(s).

Step 11 To cancel this task, click **Cancel**.

Step 12 To associate this image to the selected devices, click **Submit**.

A confirmation page appears.

Search Parameters

Each Search Parameter can be associated with an action to be performed. In this release, Search Parameters are associated with the action to delete certain files from the file system on a device.

For example, if you want to delete all files that contain **.bin** from a device, you can create a Search Parameter that states: **FileName contains .bin** and use this Precondition from the **Devices > Delete Files**.

From the Image Service Functional Overview page, click **Search Parameters**.

Note For External Directory Mode, the Search Parameters tab is called Preconditions.

The Search Parameters Functional Overview page appears showing:

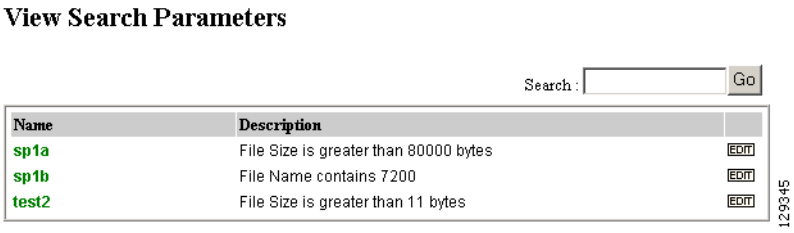
- View Search Parameters
- Create Search Parameter
- Edit Search Parameter
- Delete Search Parameters

Viewing Search Parameters

Step 1 From the Search Parameters Functional Overview page, click **View Search Parameters**.

The View Search Parameters page appears (see Figure 18-9).

Figure 18-9 View Search Parameters



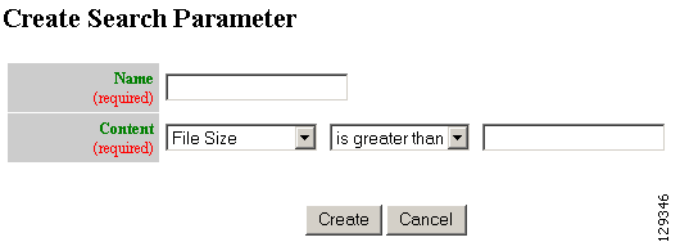
Step 2 To edit a Precondition, click **Edit** for the desired Precondition, then go to "Editing Search Parameters" section on page 18-181.

Creating Search Parameters

Step 1 From the Search Parameters Functional Overview page, click **Create Search Parameter**.

The Create Search Parameter page appears (see Figure 18-10).

Figure 18-10 Create Search Parameter



Step 2 Enter the name of this Search Parameter.

Step 3 Use the drop-down arrow in the left Content menu to select:

File Size

File Name

File Timestamp

a. For **File Size**, use the drop-down arrow in the center Content menu to select:

is greater than

is less than

is equal to

b. For **File Name**, the only choice is **contains**.

c. For File Timestamp, the only choice is **before**.

Step 4 Enter the remaining portion of the argument in the right Content field.

For example:

File Size is greater than 80,000 bytes

Step 5 To cancel this task, click **Cancel**.

Step 6 Click **Create**.

Editing Search Parameters

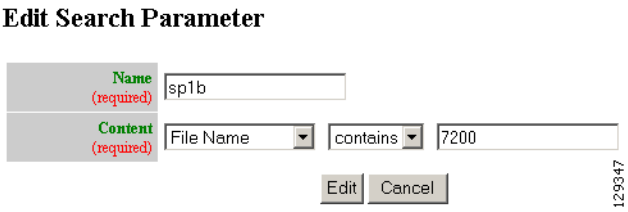
Step 1 From the Search Parameters Functional Overview page, click **Edit Search Parameter**.

The Edit Search Parameter page appears.

Step 2 Select the Search Parameter you want to edit.

The argument page for the Search Parameter appears (see Figure 18-11).

Figure 18-11 Edit Search Parameter Argument



Step 3 Edit the name or argument as required.

Step 4 To cancel this task, click **Cancel**.

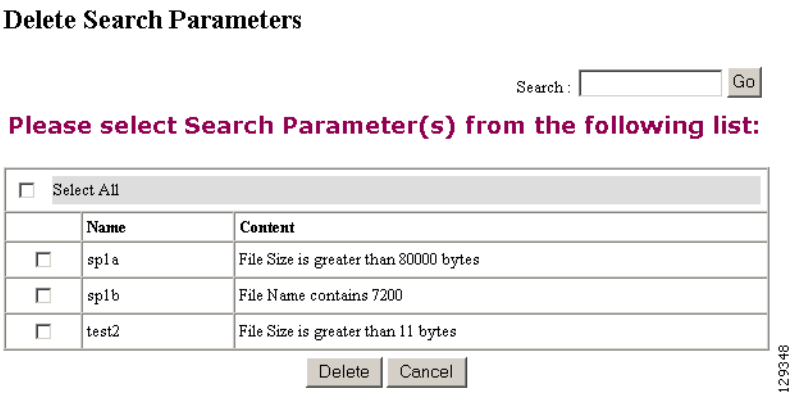
Step 5 To save your changes, click **Edit**.

Deleting Search Parameters

Step 1 From the Search Parameters Functional Overview page, click **Delete Search Parameter**.

The Delete Search Parameters page appears (see Figure 18-12).

Figure 18-12 Delete Search Parameters



Step 2 Check to select the Search Parameter(s) to delete, then click **Delete**.

Upgrade or Downgrade Cisco IOS Image

With the Image Service feature, you can not only update the Cisco IOS image on a device, you can revert back to an earlier version of the image. When you do this, the availability of Cisco Configuration Engine agents on the device might change. This means you might have to use IMGW to simulate agents to update configurations and images on the device.

Cisco Configuration Engine agents at the device-level are a function of the particular version of Cisco IOS running on that device:

- 12.0 or earlier – No Cisco Configuration Engine agents on the device
- 12.2 – Configuration Agent and Event Agent but not the Image Agent
- 12.3(3) or later – Configuration Agent, Event Agent, and Image Agent

Things to Know

- IMGW can simulate different agent types:
  - Configuration Agent only
  - Image Agent only
  - both Configuration Agent and Image Agent

  Make sure to select the correct agent for your purpose when creating IMGW devices.
- You should always have one set of the same agents running for the same device object. The common mistake when upgrading/downgrading to a different version of an image is:
  - Upgrading: after enabling a certain agent on the device, you still have an IMGW device that is simulating the same agent on the Cisco Configuration Engine, or the other way around.
  - Downgrading: a certain agent is not available on the device anymore, but the IMGW device is not simulating this agent. The next update will fail.

12.0 -> 12.2

To update an image from 12.0 to 12.2, the image needs to use IMGW to simulate both Configuration Agent and Image Agent.

Procedure

Step 1 Create a template for configuration update. This template only applies to a device when you do a configuration update.

Step 2 Create a template for image activation.

The activation template should include the boot image information. For example, if you want to copy image *c837-k9o3y6-mz.122-13.ZH2.bin* to flash and run it as the active image, the following CLI commands should be in the active template:

no boot system

boot system flash:c837-k9o3y6-mz.122-13.ZH2.bin

Step 3 Create the image for the device:

a. Setup an FTP/TFTP server.

b. Copy the image onto the FTP/TFTP server.

c. Log into the Cisco Configuration Engine, go to I**mage Service** -> **Images** -> **Create Image**.

d. Enter image information on the page or just enter **Name** and **Image Locations** on the FTP/TFTP server, then click on **Populate** to get image information.

e. Click on **Create**.

f. To verify, go to **Image Service** -> **Images** -> **View Image**, select the image and verify the image information.

Step 4 Create an IMGW device with device hop info. Make sure to select an agent type to simulate both Configuration Agent and Image Agent (see "Adding Non-agent Enabled Devices" section on page 3-31).

Step 5 Update image (see "Updating Device Images" section on page 3-67).

Step 6 To check the updating status, go to **Jobs** -> **Query Job**, click **Status** to check the job status.

Step 7 To see more debug message on the job, go to **Log Manager** -> **View Logs** and select the log to view.

Step 8 Now you should have the 12.2 image running on the device. If you want to enable Configuration Agent and Event Agent on the device, put the following CLI commands in device configuration template that you created in Step 1, then do **Update Confi**g from Cisco Configuration Engine:

cns config partial server_ipaddress port

cns event server_ipaddress port

Step 9 To verify, go to the View Device page on Cisco Configuration Engine. You should be able to see a green indicator next to this device object.

Note In order to use Configuration Agent and Event Agent to do configuration updates, you should delete the IMGW device object since it should never have two sets of the same agent for the device on the Cisco Configuration Engine.

12.0 -> 12.3(3) or later

To update image from 12.0 to 12.3(3) or later image you need to use IMGW to simulate both Configuration Agent and Image Agent.

The image update procedure is the same as 12.0 -> 12.2 except in Step 9. To enable the image agent on the device, you can also add the following line to the configuration template and update the configuration to the device:

cns image server http://*server_ipaddress*/cns/HttpMsgDispatcher status http://*server_ipaddress*/cns/HttpMsgDispatcher

Note In order to use Configuration Agent, Event Agent, and image agent to do configuration and image updates, you should delete the IMGW device object since it should never have two sets of the same agent for a device on the Cisco Configuration Engine.

12.2 -> 12.3(3) or later

There are two ways to update the image from 12.2 to 12.3(3) or later image:

1. No agents enabled on the device and use IMGW to simulate both Configuration Agent and Image Agent. The procedure is same as update from 12.0 -> 12.2.
2. Enable Event Agent and Configuration Agent on devices to update activation template and use IMGW to simulate image agent only.

Procedure

Step 1 On the device, make sure to enable Configuration and Image Agents with the following commands (it can be done from router command line or from Cisco Configuration Engine configuration update):

cns event *server_ipaddress port*

cns config partial *server_ipaddress port*

Step 2 Repeat the procedure in 12.0 -> 12.2 except in Step 4. When creating the IMGW device, make sure to select **Image Agent** for Agent Type.

Step 3 To enable the image agent on the device, you can also add the following line to configuration template and update configuration to the device:

cns image server http://server_ipaddress:http_port/cns/HttpMsgDispatcher status http://server_ipaddress:http_port/cns/HttpMsgDispatcher

Note In order to use Configuration Agent, Event Agent, and Image Agent to do configuration and image updates, you should delete the IMGW device object since it should never have two sets of the same agent for a device on the Cisco Configuration Engine.

12.3(3) or later -> 12.3(3) or later

Image upgrading from 12.3(3) or later -> 12.3(3) later images can be done with agents enabled on device. There is no need for IMGW.

Procedure

Step 1 On the device, make sure to enable the Configuration Agent with the following commands (it can be done from router command line or from Cisco Configuration Engine configuration update):

cns event ***server_ipaddress prot***

cns config partial ***server_ipaddress prot***

cns image server http://*server_ipaddress*/cns/HttpMsgDispatcher status http://*server_ipaddress*/cns/HttpMsgDispatcher

Step 2 Create a template for configuration updates.

Step 3 Create a template for image activation.

Step 4 Create an image for device:

a. Setup FTP/TFTP server.

b. Copy image on FTP/TFTP server.

c. Log into the Cisco Configuration Engine, go to **Image Service** -> **Images** -> **Create Image**.

d. Enter image information on the page or just enter **Name** and **Image Locations** on the FTP/TFTP server then click **Populate** to get image information.

e. Click on **Create**.

f. To verify, go to **Image Servic**e -> **Images** -> **View Image**, select the image and verify the image information.

Step 5 Create a device object on Cisco Configuration Engine (see "Adding Agent Enabled Devices" section on page 3-39).

Step 6 Associate the device object with an image object.

Step 7 Update image see "Updating Device Images" section on page 3-67.

Step 8 To check the updating status, go to **Jobs** -> **Query Job**, click the **Status** to check the job status.

Step 9 To see more debug messages on the job, go to **Log Manager** -> **View Logs** and select the log to view.

12.3(3) or later -> 12.2

This is the same as upgrading from 12.2 -> 12.3(3) or later images. There are several things that you should check before submitting the update:

- If you are using the second option in 12.2->12.3(3), which uses IMGW to simulate only the Image Agent, but not the Configuration Agent and Event Agent, make sure there is only Event Agent and Configuration Agent enabled on the device but no Image Agent; even though it is running 12.3(3) or later image that has all the agents. The IMGW on the server side will simulate the Image Agent.

- If there is already a device on the Cisco Configuration Engine, you only need to add an IMGW device with the same device name as device object on Cisco Configuration Engine.
- Please remove any commands in your configuration template to configuration Image Agent.

12.3(3) or later -> 12.0

Same as upgrading from 12.0 -> 12.3(3) or later image. There are several things that users should check before submit the update:

Step 1 Make sure there is no agent enabled on router even it runs 12.3(3) or later image that has all the agents. The IMGW on server side will simulate both Configuration Agent and Image Agent.

Step 2 If there is already device object on the Cisco Configuration Engine, users only need to add IMGW device with the same device name as device object on Cisco Configuration Engine.

Step 3 Please remove them if you have any command in your configuration template to configure Configuration Agent, Event Agent, or Image Agent.

Backup and Restore

This chapter describes Backup and Restore management tasks. The Backup and Restore function allows you to backup directory data (configuration templates, device and user information, and so forth) to a remote location.

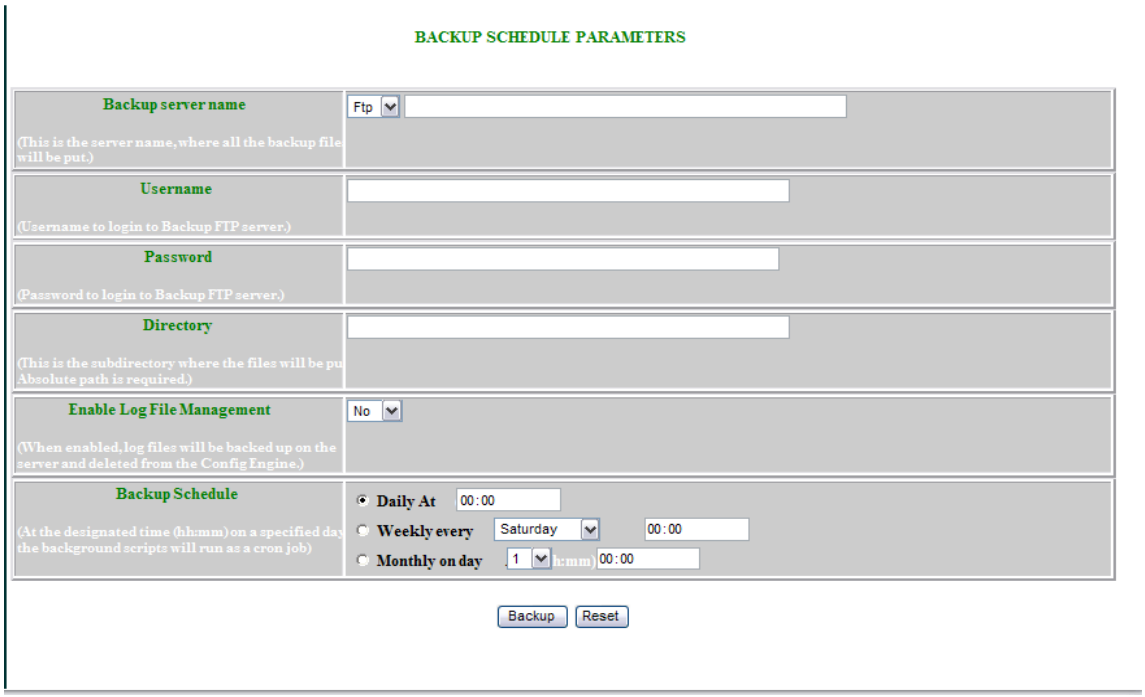
Backup Procedure

Step 1 Log into the Cisco Configuration Engine user interface.

Step 2 Go to **Tools** > **Data Manager** > **Schedule Backup**.

The backup information dialog box appears (see Figure 20-1).

Figure 20-1 Backup Schedule Parameters



Step 3 Use the drop-down arrow to select **FTP**, or **TFTP**.

Note If you select TFTP, the Username, Password, and Directory fields are disabled.

Step 4 To specify where you want the backup data to be stored, enter the FTP server name in the **FTP Server Name** field.

Note To edit or remove a scheduled backup job, enter the **crontab-e** command.

Table 20-1 shows valid values for these fields.

Table 20-1 Valid Values for Backup Schedule Parameters

Attribute	Description	Valid Values
FTP/TFTP	Select FTP or TFTP type. When you select TFTP server, the Username, Password and Directory fields are disabled because the TFTP server does not require a username and password, and all the files will go into the TFTP root directory. The file name format is ***backup-cnsce-$HOST-$DATE-$(date +%H%M).tar.gz (ex:backup-cnsce-myCE-2 0100202-1843.tar.gz)***.	From drop-down
Server name	Server name where all backup files will be put.	a-z A-Z 0-9 -(hyphen) _ (under-score) . (period)
Username	Login username for the FTP server.	a-z A-Z 0-9 -(hyphen) _ (under-score) . (period)
Password	Password for FTP server.	
Directory	Subdirectory into which all backup files will be put.	Absolute path
Enable Log File Management	determines whether files will be deleted from host system after backup.	From drop-down list
Backup Schedule	Date and time fields.	As required

Step 5 To specify the username to log into the FTP server, enter a valid username in the **Username** field.

Step 6 To specify the password to use to log into the FTP server, enter a valid value in the **Password** field.

Step 7 To specify the subdirectory where the data file is put, enter the absolute path in the **Directory** field.

Step 8 Choose whether to **Enable Log File Management**.

Step 9 To specify the backup schedule, complete the fields in the **Backup Schedule** pane.

Note The time base for the host system should be set to Coordinated Universal Time (UTC).

Step 10 To cancel this task, click **Cancel**.

Step 11 To schedule the backup operation, click **Backup**.

Data Restore Procedure

Step 1 Log in to the host system.

Step 2 Type **datarestore** at the command line, then press **Enter**.

Step 3 Provide inputs to following prompts:

Notes

Sample user inputs are shown in **bold** text.

FTP Server

```
root@i336s6 root]# datarestore
Entering Data Restore section
Type ctrl-c to exit

Enter Transfer Protocol (FTP[F] or TFTP[T]): F
Enter FTP server (hostname.domainname or IP address): 10.77.27.17
Enter username used for FTP server: root
Enter FTP password: *****
Re-enter FTP password: *****
Enter absolute pathname of backup file on FTP server: /backup.tar
```

TFTP Server

```
[root@i336s6 root]# datarestore
Entering Data Restore section
Type ctrl-c to exit

Enter Transfer Protocol (FTP[F] or TFTP[T]): T
Enter the TFTP server (hostname.domainname or IP address): 10.77.27.17
Enter pathname of backup file on the TFTP server(relative to tftp root dir):
backup.tar
```

DNS Server

```
[root@i336s6 root]# datarestore
Entering Data Restore section
Type ctrl-c to exit

Enter Transfer Protocol (FTP[F] or TFTP[T]): T
Enter the TFTP server (hostname.domainname or IP address): test.cisco.com
Enter DNS server IP address: 10.77.27.1
Enter pathname of backup file on the TFTP server(relative to tftp root dir):
backup.tar
```

Definitions

FTP: File transfer protocol.

FTP/TFTP Server: <hostname.domainname>, or IP address, of the FTP/TFTP server on which the backup file is located.

DNS Server: IP address of the DNS server. This appears when you enter a hostname instead of an IP address for the server prompt.

FTP Username: username used for FTP server.

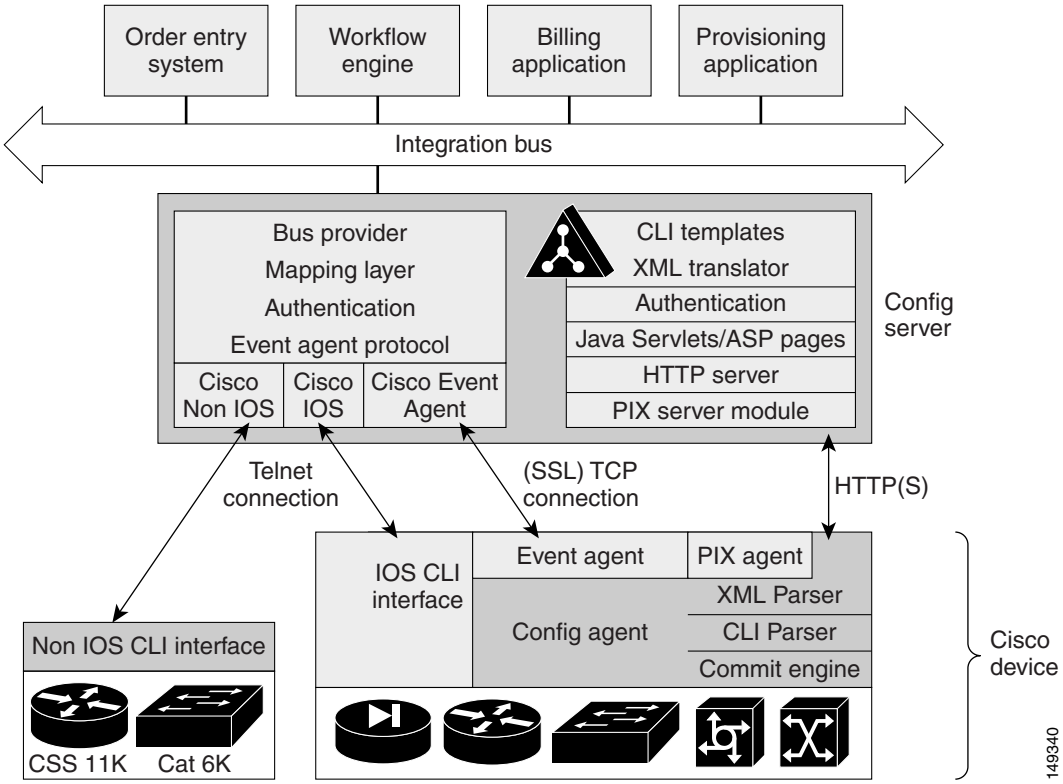
FTP Password: password used to log into the FTP server.

Absolute pathname of backup file on FTP/TFTP server: fully specified path of the backup file stored on the FTP server, or TFTP server (relative to TFTP root directory).

PIX Firewall Device Support

Cisco Configuration Engine provides configuration management and image service to Cisco PIX firewall devices (PIX device). Figure 21-1 shows a functional block diagram of Cisco Configuration Engine including the PIX device interface module.

Figure 21-1 PIX-Compatible Configuration Engine Module Interaction



Note Encryption must be enabled during setup for PIX devices to supported by Cisco Configuration Engine.

PIX Device Polls for Updates

The PIX device contacts the PIX module in the Cisco Configuration Engine to report information about itself. This occurs when the PIX starts, when any of the reported information changes, and whenever the PIX wants to check for updates. PIX sends the **DeviceDetails** message to the server. The **DeviceDetails** message provides the Cisco Configuration Engine an update of the version of software that the device is currently running. The information received in **DeviceDetails** is logged into the log file (*pix.log*) for reference.

The server responds with the **UpdateInfo** message. This message contains (optionally):

- Checksum and URL for the configuration file the PIX should be running
- Checksum and URL for the PIX image
- Checksum and URL for the PIX Device Manager (PDM) image
- URL for reporting any errors

The PIX compares the checksum in the message with the current checksum of the component concerned. In the case of configuration, it also calculates the cryptochecksum of the running configuration and compare that with the one calculated the last time when the configuration was updated from the Cisco Configuration Engine. An update is required if the checksum (or cryptochecksum) differs.

If a software/configuration update is required, the PIX sends requests on the respective URLs.

Configuration Processing

For any configuration update that is required, the PIX sends an HTTPS GET request to the returned URL. The configuration file is completely read into a local buffer before being applied. This is to prevent a connection error from leaving the PIX in a partially configured state. If there are no errors (or the *errors* attribute of the **config-data** message is *continue*) while applying the configuration commands, then the running configuration is copied to flash with the **write memory** command. All configuration files work in the *replace* mode.

Completion of configuration download by a PIX device results in a log file entry indicating the same in *pix.log*.

Note The log entry does not mean that the configuration has been successfully applied on a PIX device. It only means that the PIX device has downloaded the configuration file.

Image Processing

The **DeviceDetails** XML sent along with the initial HTTPS POST optionally has information regarding the PIX image, its version and checksum. Cisco Configuration Engine returns with the UpdateInfo XML containing image URLs and checksums based on the entries in the directory. The PIX downloads and applies images one after the other (and reload itself if required). Any error is processed as mentioned below.

Note There is no notification of successful image download because image distribution might be external to Cisco Configuration Engine and hence the PIX server cannot keep track of the same. Also, the PIX device does not provide any image upgrade successful indication.

Error Processing

All errors are reported by way of HTTPS POST to the error URL using the **ErrorList** message.

Each configuration error report (type=error, warning or info) is logged by the Cisco Configuration Engine into *pix.log*. The log file is cyclic to limit disk space usage.

Note An error occurring during configuration does not mean that the downloaded configuration is not been applied on the PIX entirely. It only means that the error mentioned in the log file has happened with respect to this particular device.

Any error or notification (type= warning, notification, informational, debugging, emergency, alert, critical and error) that occurs while retrieving the data at one of the URLs received from the Cisco Configuration Engine results in log file entries.

If a failure is encountered during the processing of any of the URLs in the UpdateInfo response from the server, the error is reported to the Error URL. Also, processing of all URLs received in the current call home is discontinued. Any further processing is deferred till the PIX calls home again.

After all the updates are successfully completed, another **DeviceDetails** message is sent to the Cisco Configuration Engine by the PIX device. Cisco Configuration Engine again sends the **UpdateInfo** and checksum. The PIX device compares the checksums and finds that no further updates are required.

Processing a DeviceDetails Request from PIX Device

The sequence of processing a DeviceDetails request from a PIX device is as follows:

1. PIX device contacts the Cisco Configuration Engine with **DeviceDetails** as XML payload by means of an HTTPS post request.
2. New PIX Configuration servlet receives request, parses XML, and retrieves DeviceID.
3. Device is authenticated.
4. Template associated with this DeviceID is processed to generate a configuration file.
5. Cnfiguration file is converted into XML format as per the PIX DTD and the file is saved (over-written in case a file is already present for this DeviceID).
6. Checksum of XML configuration file is calculated and URL noted.
7. URLs and checksums for pix image and PDM images are retrieved from image object attached with the PIX device.
8. Checksums and URLs for configuration file and various images (if the corresponding checksum differs) and the Error URL are sent to the PIX device as an HTTP response with an XML payload (UpdateInfo).
9. Device now requests for configuration/image based on the content of the UpdateInfo response.
10. If errors are encountered, information is posted to error URL.
11. Error servlet logs the errors to *pix.log*.

PIX DeviceID

The following PIX CLI decides the value of DeviceID sent by PIX in the DeviceDetails request:

[no] auto-update device-id hardware-serial | hostname | ipaddress [*if-name***] | mac-address [***if-name***] | string text**

- **auto-update device-id** command specifies the device ID to send when polling the Management server.
- **no auto-update device-id** command resets the device ID to the default of hostname.
- **hardware-serial** option uses the PIX serial number.
- **hostname** option uses the PIX host name.
- **ipaddress** option uses the IP address of the interface with the name **if-name**.

  If the interface name is not specified, it uses the IP address of the interface used to communicate with the remote management server.
- **mac-address** option uses the MAC address of the interface with the name *if-name*.

  If the interface name is not specified, it uses the MAC address of the interface used to communicate with the remote management server.
- **string** option uses the specified *text*.

  The text can not contain white space or the characters ‘, “, <, >, & and ?.

Note Since the DeviceID provided by the PIX is internally mapped to ConfigID and EventID in the Cisco Configuration Engine, it only supports hyphen (-), underscore (_), period (.) and alphanumeric characters.

Security Considerations

Since PIX devices are firewall devices and configuration information is vital, transport of this information is made secure by the use of SSL.

HTTPS has been enforced as the transport protocol between PIX devices and Cisco Configuration Engine under all circumstances. **DeviceDetails**, **Update Info**, **ErrorInfo** and configuration files are transported only using HTTPS. The authorization mechanism used in Configuration Service has been leveraged in the PIX server module. The URLs supplied by you towards PDM/pix-image can use HTTP or HTTPS.

PIX Device Polling Setup

PIX devices can be configured to poll the Cisco Configuration Engine at regular intervals for configuration or image updates. This entry has to be made by you on the PIX device itself. Details are available from PIX device documentation. CLI format for the same is as follows:

Usage: auto-update device-id hardware-serial | hostname |

ipaddress [<if_name>] | mac-address [<if_name>] | string <text>

no auto-update device-id

auto-update poll-period <poll-period> [<retry-count>

[<retry-period>]]

no auto-update poll-period

auto-update server <url> [verify-certificate]

no auto-update server

auto-update timeout <period>

no auto-update timeout

Example:

```
auto-update device-id string myPIXDevice
auto-update poll-period 120
auto-update server https://********@cns-ie2100/cns/PIXConfig
```

The URI to be polled on the Cisco Configuration Engine is:

/cns/PIXConfig

The **auto-update poll-period** command specifies how often to poll the Management server for configuration or image updates. The *poll-period* parameter specifies how often (in minutes) to check for an update. The default is 720 (12 hours). The *retry-count* option specifies how many times to try re-connecting to the server if the first attempt fails. The default is 0. The *retry-period* option specifies how long to wait (in minutes) between retries. The default is 5.

The **no auto-update poll-period** command resets the poll period to the default.

Also, you must to map the hostname of the server on the PIX device with its IP address. You can do this by using the *name* command as follows:

pixfirewall# **conf t**

pixfirewall(config)# **name <ip_address of the server> <hostname of the server>**

Configuration and Restrictions

PIX compatibility module is set up along with Configuration Service during the initial setup of the system. You need not do anything specifically to enable PIX compatibility.

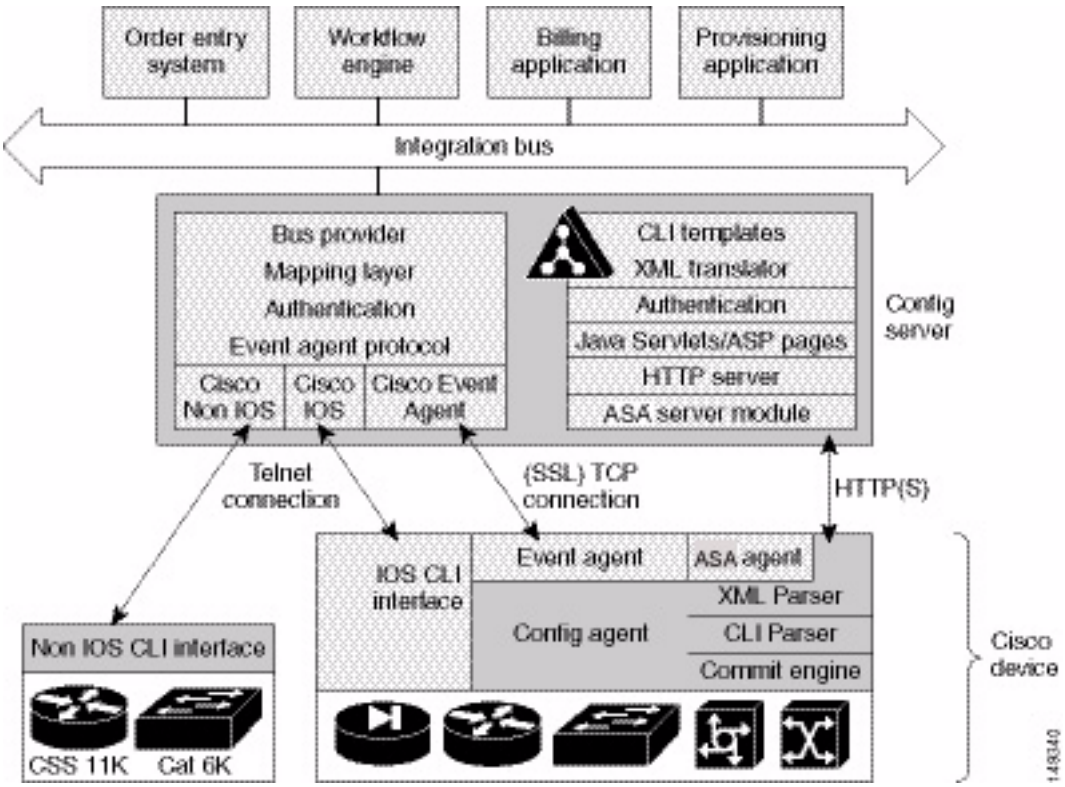
PIX devices with **software versions of 6.2.1 and higher** are supported by Cisco Configuration Engine (auto-update from PIX device side was introduced in this version). All PIX hardware platforms that run software version 6.2.1 or higher will be supported.

The configuration files will be generated with options config-action= **replace** and errors=**revert**. No other options are supported.

ASA Firewall Device Support

Cisco Configuration Engine provides configuration management and image service to Cisco Adaptive Security Appliance devices (ASA device). Figure 22-1 shows a functional block diagram of Cisco Configuration Engine including the ASA device interface module.

Figure 22-1 ASA-Compatible Configuration Engine Module Interaction



Note Encryption must be enabled during setup for ASA devices to supported by Cisco Configuration Engine.

ASA Device Polls for Updates

The ASA device contacts the ASA module in the Cisco Configuration Engine to report information about itself. This occurs when the ASA starts, when any of the reported information changes, and whenever the ASA wants to check for updates. ASA sends the **DeviceDetails** message to the server. The **DeviceDetails** provides the Cisco Configuration Engine with an update on the version of the software that the device is currently running. The information received in the **DeviceDetails** message is logged into the log file (*asa.log*) for reference.

The server responds with the **UpdateInfo** message. This message contains (optionally):

- Checksum and URL for the configuration file the ASA should be running
- Checksum and URL for the ASA image
- Checksum and URL for the ASA Device Manager (ASDM) image
- URL for reporting any errors

The ASA compares the checksum in the message with the current checksum of the component concerned. In the case of configuration, it also calculates the cryptochecksum of the running configuration and compares that with the one calculated last time when the configuration was updated from the Cisco Configuration Engine. An update is required if the checksum (or cryptochecksum) differs.

If a software/configuration update is required, the ASA sends requests on the respective URLs.

Configuration Processing

For any configuration update that is required, the ASA sends an HTTPS GET request to the returned URL. The configuration file is completely read into a local buffer before being applied. This message is used to prevent a connection error from leaving the ASA in a partially configured state. If there are no errors (or the *errors* attribute of the **config-data** message is *continue*) while applying the configuration commands, the running configuration is copied to flash with the write memory command. All configuration files work in the *replace* mode.

Completion of configuration download by a ASA device results in a log file entry indicating the same in *asa.log*.

Note The log entry does not mean that the configuration is successfully applied on a ASA device. It only means that the ASA device has downloaded the configuration file.

Image Processing

The **DeviceDetails** XML message sent along with the initial HTTPS POST optionally has information regarding the ASA image, its version, and checksum. Cisco Configuration Engine returns with the UpdateInfo XML containing image URLs and checksums based on the entries in the directory. The ASA downloads and applies images one after the other (and reloads itself if required). Any error is processed as mentioned next.

Note There is no notification for the successful image download because the image distribution can be external to Cisco Configuration Engine and hence, the ASA server cannot keep track of the same. Also, the ASA device does not provide any image upgrade successful indication.

Error Processing

All errors are reported by way of HTTPS POST to the error URL using the **ErrorList** message.

Each configuration error report (type=error, warning or info) is logged by the Cisco Configuration Engine into *asa.log*. The log file is cyclic to limit disk space usage.

Note An error occurring during configuration does not mean that the downloaded configuration is not applied on the ASA entirely. It only means that the error mentioned in the log file has happened with respect to this particular device.

Any error or notification (type= warning, notification, informational, debugging, emergency, alert, critical, and error) that occurs while retrieving the data at one of the URLs received from the Cisco Configuration Engine results in log file entries.

If a failure is encountered during the processing of any of the URLs in the UpdateInfo response from the server, the error is reported to the Error URL. Also, processing of all URLs received in the current call home is discontinued. Any further processing is deferred till the ASA calls home again.

After all the updates are successfully completed, another **DeviceDetails** message is sent to the Cisco Configuration Engine by the ASA device. Cisco Configuration Engine again sends the **UpdateInfo** and checksum. The ASA device compares the checksums and finds that no further updates are required.

Processing a DeviceDetails Request from ASA Device

The sequence of processing a DeviceDetails request from a ASA device is as follows:

1. ASA device contacts the Cisco Configuration Engine with the **DeviceDetails** message as XML payload by means of an HTTPS post request.
2. New ASA Configuration servlet receives request, parses XML, and retrieves DeviceID.
3. Device is authenticated.
4. Template associated with this DeviceID is processed to generate a configuration file.
5. Configuration file is converted into XML format as per the ASA DTD and the file is saved (over-written in case a file is already present for this DeviceID).
6. Checksum of XML configuration file is calculated and URL noted.
7. URLs and checksums for ASA image and PDM images are retrieved from image object attached with the ASA device.
8. Checksums and URLs for configuration files and various images (if the corresponding checksum differs) and the Error URL are sent to the ASA device as an HTTP response with an XML payload (UpdateInfo).
9. Device now requests for configuration/image based on the content of the UpdateInfo response.
10. Errors are encountered, information is posted to error URL.

11. The error servlet logs the errors to *asa.log*.

ASA DeviceID

The following ASA CLI decides the value of the DeviceID sent by the ASA in the DeviceDetails request:

[no] auto-update device-id hardware-serial | hostname | ipaddress [*if-name***] | mac-address [***if-name***] | string text**

- **auto-update device-id** command specifies the device ID to send when polling the Management server.
- **no auto-update device-id** command resets the device ID to the default of hostname.
- **hardware-serial** option uses the ASA serial number.
- **hostname** option uses the ASA host name.
- **ipaddress** option uses the IP address of the interface with the name **if-name**.

  If the interface name is not specified, it uses the IP address of the interface used to communicate with the remote management server.
- **mac-address** option uses the MAC address of the interface with the name *if-name*.

  If the interface name is not specified, it uses the MAC address of the interface used to communicate with the remote management server.
- **string text** option uses the specified *text*.

  The text can not contain white space or the characters ‘, “, <, >, &, and ?.

Note Since the DeviceID provided by the ASA is internally mapped to ConfigID and EventID in the Cisco Configuration Engine, it only supports hyphen (-), underscore (_), period (.) and alphanumeric characters.

Security Considerations

Because ASA devices are firewall devices and the configuration information is vital, the information is transported securely by using SSL.

HTTPS is enforced as the transport protocol between ASA devices and Cisco Configuration Engine under all circumstances. **DeviceDetails**, **Update Info**, **ErrorInfo,** and configuration files are transported using only HTTPS. The authorization mechanism used in the Configuration Service is leveraged in the ASA server module. The URLs supplied by you toward the ASDM/ASA-image can use HTTP or HTTPS.

ASA Device Polling Setup

ASA devices can be configured to poll the Cisco Configuration Engine at regular intervals for configuration or image updates. This entry has to be made by you on the ASA device itself. Details are available from ASA device documentation. CLI format for the same is as follows:

Usage: auto-update device-id hardware-serial | hostname |

ipaddress [<if_name>] | mac-address [<if_name>] | string <text>

no auto-update device-id

auto-update poll-period <poll-period> [<retry-count>

[<retry-period>]]

no auto-update poll-period

auto-update server <url> [verify-certificate]

no auto-update server

auto-update timeout <period>

no auto-update timeout

Example:

```
auto-update device-id string myASADevice
auto-update poll-period 120
auto-update server https://********@cns-ie2100/cns/ASAConfig
```

The URI to be polled on the Cisco Configuration Engine is:

/cns/ASAConfig

The **auto-update poll-period** command specifies how often to poll the Management server for configuration or image updates. The *poll-period* parameter specifies how often (in minutes) to check for an update. The default is 720 (12 hours). The *retry-count* option specifies how many times to try re-connecting to the server if the first attempt fails. The default is 0. The *retry-period* option specifies how long to wait (in minutes) between retries. The default is 5.

The **no auto-update poll-period** command resets the poll period to the default.

Also, you must map the hostname of the server on the ASA device with its IP address. You can do this by using the *name* command as follows:

asafirewall# **conf t**

asafirewall(config)# **name <ip_address of the server> <hostname of the server>**

Configuration and Restrictions

The ASA compatibility module is set up along with Configuration Service during the initial setup of the system. You need not do anything specifically to enable ASA compatibility.

ASA devices with **software versions of 8.2 and higher** are supported by Cisco Configuration Engine (auto-update from the ASA device side was introduced in this version). All ASA hardware platforms that run software version 8.2 or higher will be supported.

The configuration files will be generated with options config-action= **replace** and errors=**revert**. No other options are supported.

Following are the different type of configuration-actions.

- Replace—Specifies that the current configuration should be cleared before applying the new configuration.
- Merge—Merges the current configuration with the new configuration file.

Following are the different type of error-actions.

- Continue—Continues with applying the new configuration, even if there is a configuration error.

- Revert—Reverts the old configuration from the flash without rebooting when there is a configuration error.
- Stop—Stops reading the rest of the configuration when a command causes an error.

IMGW Device Module Development Toolkit

The Intelligent Modular Gateway (IMGW) device module development toolkit clearly defines the southbound interface of IMGW and provides a registration utility to allow you to register plug-in device modules into IMGW after the device module is installed onto the Cisco Configuration Engine.

This chapter analyzes the requirements of the IMGW device module development toolkit and describes the functionality that is offered by this toolkit.

Note You can also implement the device module in either shell scripts or Linux executables as long as the device module conforms to IMGW southbound interface.

User Types

This toolkit is oriented to three types of users:

- *Plug-in Developer*—responsible for developing the device module that complies with the IMGW southbound interface defined in this toolkit
- *System Administrator*—responsible for the following:
  - Plug the device module into and out of the Cisco Configuration Engine
  - Register and de-register the plug-in device module
  - Update the device module on the Cisco Configuration Engine
- *Network Operator*—configures the device through the plug-in device module

Toolkit Usage

There are three common usages of this toolkit:

- Plug a device module into Cisco Configuration Engine and configure devices using the device module.
- Update a device module on the Cisco Configuration Engine and configure devices through the modified device module.
- Unplug a device module from the Cisco Configuration Engine.

Plug Device Module Into Cisco Configuration Engine

Step 1 The *Plug-in Developer* develops a device module conforming to the IMGW southbound interface defined in this toolkit to handle the given device type.

For information about the device module syntax, see "IMGW Southbound Interface" section on page 23-206.

Step 2 The *System Administrator* installs the device module onto Cisco Configuration Engine.

Step 3 The *System Administrator* runs the registration utility to register the device module into IMGW.

Step 4 The *Network Operator* configures devices through the device module.

Update Device Module on Cisco Configuration Engine

Step 1 The *Plug-in Developer* provides a new version of the device module.

Step 2 The *System Administrator* runs the registration utility to de-register the device module from IMGW.

If the device module you want to update is not registered, skip this step

Step 3 The *System Administrator* updates the device module with the new version on Cisco Configuration Engine.

Step 4 The *System Administrator* runs registration utility to register the updated device module into IMGW.

Step 5 The *Network Operator* configures devices through modified device module.

Unplug Device Module from Cisco Configuration Engine

Step 1 The *System Administrator* runs the registration utility to de-register the plug-in device module from IMGW.

Step 2 The *System Administrator* uninstalls the plug-in device module from Cisco Configuration Engine.

IMGW Southbound Interface

When a command execution or a configuration update event is received by IMGW runtime, it will first retrieve device type information from the device information database. If the device module corresponding to device type and operation type (**CONFIG_UPLOAD** or **CONFIG_DOWNLOAD**) is registered, IMGW runtime forks a process to execute the proper plug-in program and pass the parameter list to the plug-in program.

The initial mapping information from the <*device type, operation type*> pair to the plug-in program is read from a configuration file into memory upon start up. When IMGW is running, the system administrator can still add, remove, or update the entries of mapping information by way of the toolkit registration utility.

The *System Administrator* can modify only the entries for non-legacy device modules. This restriction is enforced by IMGW runtime.

User Designed Device Module Specifications

A user-defined device module must conform to the IMGW southbound interface as specified in this section.

Config Event

<plug-in program> *<temp_logfile_name> <logging_level> <device_id> <action_type> <warning_logfile_name> <error_logfile_name> <hop_information_string> <configuration_file_name> <persistence> <operation_timeout_value> <prompt_timeout_value>.*

Exec Event

<plug-in program> *<temp_logfile_name> <logging_level> <device_id> <action_type> <hop_information_string> <command_to_be_executed> <command_arguments> <exec_response_logfile_name> <operation_timeout_value> <prompt_timeout_value>.*

Hop Test

<plug-in program> *<temp_logfile_name> <logging_level> <device_id> <action_type> <hop_information_string> <operation_timeout_value> <prompt_timeout_value>.*

Note All files specified for the IMGW southbound interface are managed by IMGW runtime and their file names are absolute path names.

Parameter Descriptions

Plug-in Program: The plug-in program that is executed in the child process forked by IMGW runtime. The system administrator gives this information to IMGW runtime during registration.

temp_logfile_name: The full path to the device module temporary log file, which should be used by the device module to log the processing history of one instance of operation (configuration download, command execution or hop test). This file is by default located at */tmp* directory on the Cisco Configuration Engine. After the plug-in program exits, IMGW runtime puts the content of this file into a centralized log file named */opt/CSCOimgw/bin/IMGW-DEVMOD_LOG* for debugging purpose, then unlinks this file.

logging_level: It could be verbose, error, or silent. This flag can be set up by running setup command on the host system. It is recommended that the device module log information into the file *<temp_logfile_name>* based on the specified logging level.

device_id: The identification of the device that is processed by the device module. It is passed in by the *cisco.mgmt.cns.config.load* or *cisco.mgmt.cns.exec.cmd* event.

action_type: It could be **config**, **exec**, or **hoptest**. Action type **config** notifies the device module to update the device configuration. Action type **exec** notifies the device module to execute a command on the device. Action type **hoptest** notifies the device module to test if the device is reachable by way of the hop information provided in <*hop_information_string*>. The device module should do the proper operation in response to this flag.

warning_logfile_name: The full path to the file that is used by the device module to log all warning messages and its corresponding configuration commands line numbers. This parameter is supplied by IMGW runtime only when the action type is **config** because the information in this file is only used to generate the response message to the *cisco.mgmt.cns.config.load* event if the configure succeeds with warnings. In order for the IMGW runtime to generate the proper response message, each warning message should begin a new line and be prefixed with the string of **LINE <***line number of the configuration command that causes the warning message***>:** An example of the warning file is as follows:

```
LINE 3: The interface has already been removed
.
.
.
LINE 7: The interface already exists.
```

The location of this file is under */tmp* on the host system. After the plug-in program exits, IMGW runtime puts the content of this file into the response event payload, then immediately unlinks this file.

error_logfile_name: The full path to the file that is used by the device module to log the occurrences of the error messages and their corresponding configuration command line numbers. This parameter is supplied by IMGW runtime only when the action type is **config** because the information in this file is only used to generate the response message to the *cisco.mgmt.cns.config.load* event if the configure fails. In order for the IMGW runtime to generate the proper response message, each error message should begin a new line and be prefixed with the string of **LINE <***line number of the configuration command that causes the error message***>**.

An example of the error file is as follows:

```
LINE 3: % Invalid input detected at
LINE 7: % Incomplete command
.
.
.
LINE 12: % The interface already exists
```

The location of this file is under */tmp* on the host system. After the plug-in program exits, IMGW runtime puts the content of this file into the response event payload, then immediately unlinks this file.

exec_response_logfile_name: The full path to the file that is used to log the output of command execution on the device. It is supplied by IMGW runtime only when the action type is **exec** and its location is under */tmp* on the host system. After the plug-in program exits, IMGW runtime puts the content of this file into the response event payload, then immediately unlinks this file.

hop_information_string: The string used to store the access information of the device. It is the string concatenation of all individual hop information of the device in order. An example the hop information and its <*hop_information_string*> are as follows:

Hop type	IP address	Port	Username	Password
IOS_LOGIN	172.29.145.45		Admin	Cisco
IOS_EN			Lab	Lab

The corresponding <*hop_information_string*> should be as follows:

```
"IOS_LOGIN" "172.29.145.45" " " "Admin" "Cisco" "IOS_EN" " " " " "Lab" "Lab"
```

Note For those fields of hop information with null value, IMGW runtime automatically adds a space before passing it to the child process.

command_to_be_executed: The command to be executed on the device. It is supplied by IMGW runtime only when the action type is **exec**.

command_arguments: The arguments of the command to be executed on the device. It is supplied by IMGW runtime only when the action type is **exec**.

configuration_file_name: The full path to the configuration file which will be downloaded onto the device. It is supplied by IMGW runtime only when the action type is **config** and its location is under */tmp* on the host system. After the plug-in program exits, IMGW runtime immediately unlinks this file.

persistence: y or **n**. The value **y** means the configuration needs to be written into non-volatile storage. It is supplied by IMGW runtime only when the action type is **config**. This option is dependent on the device type. This means the device module can ignore it if the device type does not support it.

operation_timeout_value: The maximum time period allowed to execute a command on the device. This parameter is now used by Expect scripts in IMGW legacy device module for IOS, CatOS, CatIOS, PIX, CSS and CE devices. A user-defined device module can ignore this parameter if it does not use it.

prompt_timeout_value: The maximum time period allowed to wait for the next prompt during login session to the device. This parameter is now used by Expect scripts in IMGW legacy device module for IOS, CatOS, CatIOS, PIX, CSS and CE devices. A user-defined device module can ignore this parameter if it does not use it.

Exit Codes

When the forked process (in which the plug-in program is executed) exits, the following exit codes are expected by IMGW runtime from the forked process:

config event:

0 – Download succeeds

1 – Download fails

2 – Download succeeds but with warning messages

Exec Event:

0 – Command execution succeeds

1 – Command execution fails

Hop Test:

0 – Hop test succeeds

1 – Hop test fails

How to Develop Plug-in Device Module

This toolkit allows the *Plug-in Developer* to use any implementation to realize the plug-in device module as long as the device module complies with IMGW southbound interface specified in "IMGW Southbound Interface" section on page 23-206.

This toolkit also provides sample code (see Toolkit Usage, page 23-205) in Perl plus Expect scripts as well as inline comments to help beginners to understand the workflow of the plug-in device module.

The plug-in device module should render three basic functions:

- Device configuration update
- Command execution
- Hop test

The first two functions are in response to the *cisco.mgmt.cns.config.load* and *cisco.mgmt.cns.exec.cmd* events respectively. The last one is an internal routine operation required by IMGW runtime and is transparent to network operators.

After IMGW runtime spawns a child process to execute the plug-in program, the corresponding device module should read the action type from the parameter list. If the action type is:

- **config** – device module should do device a configuration update.
- **exec** – device module should do a command execution.
- **hoptest –** device module should do hop test.

Development Guidelines

The following subsections describe the processes associated with each function.

Note The subject of actions in the subsections below is the plug-in device module.

Device Configuration Update

1. Access the device by way of the <*hop_information_string*>.
2. Download the configuration file named after <*configuration_file_name* > onto the device.
3. If the above download operation succeeds, the <*persistence*> is set to **y** and the device supports this option, then write the configuration to non-volatile storage.
4. Write all warning messages prompted by the device and their corresponding configuration commands line numbers into the file named after <*warning_logfile_name*> in the specified format (see "Parameter Descriptions" section on page 23-207). The content of this file will be part of the payload of the response event if the download succeeds but with warning messages.
5. Write all error messages prompted by the device and their corresponding configuration commands' line numbers into the file named after <*error_logfile_name*> in the specified format (see "Parameter Descriptions" section on page 23-207). The first error message and its corresponding configuration command line number will be part of the payload of the response event if the download fails.
6. Based on the <*logging_level*>, selectively redirect the processing history into the file named after <*temp_logfile_name*> for debugging purpose during the whole procedure.

7. Exit with proper exit code to return control to IMGW runtime. See "Exit Codes" section on page 23-209 to get the definition of exit codes.

Command Execution

1. Access the device by way of the <*hop_information_string*>.
2. Execute on the device the <*command_to_be_executed*> with the <*command_arguments*>.
3. Capture all output from the command execution into the file named after <*exec_response_logfile_name*>. The content of this file will be part of the payload of the response event.
4. Based on the <*logging_level*>, selectively redirect the processing history into the file named after <*temp_logfile_name*> for debugging purpose during the whole procedure.
5. Exit with proper exit code to return control to IMGW runtime. See "Exit Codes" section on page 23-209 to get the definition of exit codes.

Hop Test

1. Access the device by way of the <*hop_information_string*>.
2. Based on the <*logging_level*>, selectively redirect the processing history into the file named after <*temp_logfile_name*> for debugging purpose during the whole procedure.
3. Exit with proper exit code to return control to IMGW runtime. See "Exit Codes" section on page 23-209 to get the definition of exit codes.

Installing Plug-in Device Module

The *System Administrator* is required to take charge of the install/uninstall. He/She should make sure the installation is successful before calling the registration utility.

The System Administrator should install all plug-in device modules into the reserved file directory of */opt/ConfigEngine/CSCOimgw/plugin-modules* with one subdirectory per device module. For example, install the device module for MGX into */opt/ConfigEngine/CSCOimgw/plugin-modules/MGX* while install the one for NT into */opt/ConfigEngine/CSCOimgw/plugin-modules/NT*.

The *System Administrator* should only operate within the device module installation directory to set/remove the running environment of the module. The installation activities should not affect the running environment of other components on the Cisco Configuration Engine.

Registering Plug-in Device Module

The *System Administrator* must provide the device type and the full path to the plug-in program when registering a device module. IMGW runtime does not check the integrity of this information. It is responsibility of the *System Administrator* to make sure the information is correct.

This toolkit provides a dynamic registration utility to the system administrator, which allows the *System Administrator* to plug the device module into and out of IMGW seamlessly without tearing down IMGW runtime. Therefore, the services irrelevant to the device module that is being registered/de-registered will not be affected. However, this might not be the case for other services.

For example, at the time you issue the de-register command on device module *x*, the events related to *x* that are still queued in event bus might get failure responses from IMGW.

Caution It is HIGHLY RECOMMENDED that the *System Administrator* notify all *Network Operators* of the upcoming registration activities so that *Network Operators* have a chance to stop beforehand any relevant operation.

End User Interface

The end user interface of IMGW device module development toolkit consists of IMGW southbound interface and the command line registration utility.

Configuration and Restrictions

This toolkit does not put a restriction on the maximum number of plug-in device modules that can be put into IMGW.

Device Module Restrictions

- The device module must be able to run on the Linux platform.
- If the executable of the device module is a C++ binary file, it must utilize the glib that exists on Cisco Configuration Engine where applicable.
- If the executable of the device module is a java class, it must run in the existing JVM of Cisco Configuration Engine.
- If the device module includes Perl and/or Expect scripts, the scripts should use the Perl and/or Expect interpreters that exist on Cisco Configuration Engine.

Registration Utility Restriction

The *System Administrator* is not allowed to register/de-register IMGW legacy device module. Sometimes users might want to modify one of the legacy device modules to do upload/download operation on CatOS, CatIOS, PIX, CSS, CE or IOS devices to meet their specific needs. In this case, they can only modify their own copy of the legacy device module, associate a different device type name to the modified device module and register the device module into IMGW.

Troubleshooting

This appendix provides troubleshooting information. It contains information about:

- Contacting Cisco TAC
- Checking the Version Number of Cisco Configuration Engine
- Cannot Log in to the System
- System Cannot Connect to the Network
- Cannot Connect to the System Using a Web Browser
- Problems Connecting to the System with Secure Shell
- Cannot Connect to the System Using Telnet
- Backup and Restore Not Working Properly
- Cannot Back Up Jobs
- Using the cns-send and cns-listen Commands

Note For additional troubleshooting information, see the *Troubleshooting Guide for Cisco Configuration Engine*.

Contacting Cisco TAC

In some of the sections, you might be advised to contact the Cisco Technical Assistance Center (TAC) for assistance. You can obtain TAC assistance online at http://www.cisco.com/tac.

Checking the Version Number of Cisco Configuration Engine

To check the version number of the Cisco Configuration Engine software, do one of the following:

- Start the Cisco Configuration Engine application, and look for the version number in the displayed login screen.
- Use the **version** command. This command is located in the *cd $CISCO_CE_INSTALL_ROOT/CSCOcnsie/bin* directory.

Cannot Log in to the System

Problem You cannot log in to the system.

Possible Cause This problem could occur for one of the following reasons:

- You did not run the Setup program to create the initial system configuration.
- You lost all of the user account passwords.

Solution To resolve this problem, follow these steps:

Step 1 If you did not run the Setup program, run the Setup program as described in the *Cisco Configuration Engine Solaris Installation & Configuration Guide, 2.0*.

Step 2 If you do not know the passwords for the system user accounts, reconfigure the system to create a new user account.

Step 3 If you still cannot log in to the system, contact the Cisco Technical Assistance Center (TAC) for assistance.

System Cannot Connect to the Network

Problem The system cannot connect to the network.

Possible Cause This problem could occur for the following reasons:

- The network cable is not connected to an Ethernet port.
- The Ethernet interface is disabled or misconfigured.
- The system is configured correctly, but the network is down or misconfigured.
- The system is not configured correctly.

Solution To resolve this problem, follow these steps:

Step 1 Verify that the network cable is connected to an Ethernet port and that the Link light is on.

- If the network cable is not connected, connect it.
- If the network cable is connected but the Link light is not on, check these probable causes:
  - The network cable is faulty.
  - The network cable is the wrong type (for example, a crossover type is used, instead of the required straight-through type).
  - The port on the default gateway to which the system connects is down.

Step 2 If you still cannot connect to the network, use the **ping** command to perform the following tests:

a. Try to connect to a well-known host on the network. A DNS server is a good target host.

If the **ping** command can reach the well-known host, the system is connected to the network. If it cannot connect to the host, the problem is with the network configuration or the host. Contact your network administrator for assistance.

b. If the **ping** command cannot reach the well-known host, try to reach another host on the same subnet as the system.

   If the **ping** command can reach a host on the same subnet, but cannot reach a host on a different subnet, the default gateway is probably down or misconfigured.

Step 3 If the **ping** command cannot reach any hosts, use the **ifconfig** command to determine whether the Ethernet interface is disabled or misconfigured.

If the Ethernet interface is disabled, enable it. If it is misconfigured, configure it correctly.

Step 4 If the interface is enabled and correctly configured but you still cannot connect to the network, ensure that all network setting are configured correctly. Run the Setup program again by entering the **setup** command in the shell prompt.

Note You cannot run the Setup program a second time by logging in as **setup**. For security reasons, the account is disabled after it is used once successfully.

Step 5 Contact your network administrator to verify that there are no conditions on the network that prevent the system from connecting to the network.

Step 6 If no conditions are preventing the system from connecting to the network, contact the Cisco TAC for assistance.

Cannot Connect to the System Using a Web Browser

Problem You cannot connect to the system by entering its IP address in a web browser.

Possible Cause This problem could occur for the following reasons:

- The system cannot connect to the network.
- Encryption is enabled (plain text is disabled).
- The HTTP service is not running.

Solution To resolve this problem, follow these steps:

Step 1 Make sure that the system can connect to the network.

If it cannot connect to the network, see the "System Cannot Connect to the Network" section on page A-2 for possible resolution.

Step 2 Try to connect to the system by using a web browser.

If encryption is enabled:

- Use **https://**… to connect.
- Verify that the certificate is correct.

Step 3 If you still cannot connect, stop and start the web server by entering the following commands:

```
/etc/rc.d/init.d/httpd stop
/etc/rc.d/init.d/httpd start
```

If the LDAP directory contains thousands of devices, restart and wait 20 minutes.

Step 4 Repeat Step 2.

Step 5 If you cannot connect, restart the system.

If the LDAP directory contains thousands of devices, restart and wait 20 minutes.

Step 6 If you still cannot connect to the system, contact the Cisco TAC for assistance.

Problems Connecting to the System with Secure Shell

Problem When connecting to the system using Secure Shell (SSH), you experience one of these problems:

- You cannot connect to the system.
- The system is extremely slow, even though it is connected to the network.
- The system cannot correctly process requests from management applications.

Possible Cause The system cannot obtain DNS services from the network.

Solution To resolve this problem, follow these steps. Connect to the console if you cannot connect by using SSH.

Step 1 Do one of the following:

- Set up the name servers properly by editing the */etc/resolv.conf* file.
- Re-execute **Setup**.

Step 2 Verify that the system can obtain Domain Name System (DNS) services from the network by entering the following command:

```
# host <dns-name>
```

where *<dns-name>* is the DNS name of a host on the network that is registered in DNS. When you enter this command, it responds with the IP address of the host.

If the system cannot resolve DNS names to IP addresses, the DNS server is not working properly.

Step 3 Resolve the network DNS problem.

Step 4 If the system can resolve DNS names to IP addresses but you still cannot connect to the system using SSH, contact the Cisco TAC for assistance.

Cannot Connect to the System Using Telnet

Problem You cannot connect to the system by using Telnet even though the system is connected to the network.

Possible Cause This problem could occur if the Telnet service is disabled on the system.

Solution To resolve this problem, use SSH to connect to the system.

Backup and Restore Not Working Properly

Problem Backup and restore is not working properly.

Possible Cause This problem could occur for the following reasons:

- The time base for the host system is not set to the UTC time zone.
- The time has changed.
- The cron job has not started.

Solution To resolve this problem, follow these steps:

Step 1 Connect to the console if you cannot connect using SSH.

Step 2 Log in to the host system as root.

Step 3 To determine whether the time is correct, enter the following command:

```
# date
```

Step 4 To determine the state of the cron job, enter the following command:

```
# /etc/rc.d/init.d/crond restart
```

Example:

```
# /etc/rc.d/init.d/crond restart
Stopping cron daemon:                                      [  OK  ]
Starting cron daemon:                                      [  OK  ]
#
```

Cannot Back Up Jobs

Problem Cannot back up jobs.

Possible Cause The **crontab** command is used to schedule backup jobs. This command requires space in the */var* partition to execute. If the */var* partition is full, the **crontab** command fails to execute, which causes backup job failure.

Solution To resolve this problem, clean up the */var* partition on the system (move some files to *the /home/* directory). Then resubmit the backup job from the Cisco Configuration Engine user interface.

Using the cns-send and cns-listen Commands

Use the **cns-send** and **cns-listen** commands to send and receive test messages to the event gateway in the Cisco Configuration Engine. These commands are located in the */opt/CSCOcnsie/tools* directory.

cns-send

The syntax for the cns-send command is:

cns-send -version

or

cns-send [-service <*service***>] [-network <***network***>] [-daemon <***daemon***>] [-file <***filename***>] <***subject***> [<***message***>]**

Syntax Description

-version	Outputs the version of cns-send.
-service <*service*>	(Optional) The port number (default: 7500).
-network <*network***>**	(Optional) Network interface (in local machine) where messages are sent.
-daemon <*daemon***>**	(Optional) Internal port of application to the rvd daemon (default: 7500).
-file <*filename***>**	(Optional) Filename containing the XML-message. The filename can be sent instead of individual subject/messages.
<*subject***>**	Subject name of the message.
<*message***>**	(Optional) Message in the message field.

To use the cns-send command, follow these steps:

Step 1 Log in to the host system as root.

Step 2 Change directories to **/opt/CSCOcnsie/tools**.

Step 3 Type **./cns-send -file <***filename***> <***subject***>**

Note The cns-send command sends messages in the opaque data format.

cns-listen

The syntax for the cns-listen command is:

cns-listen -version

or

cns-listen [-service <*service***>] [-network <***network***>] [-daemon <***daemon***>] <***subject_list***>**

Syntax Description

-version	Outputs the version of cns-listen.
-service <*service*>	(Optional) The port number (default: 7500).
-network <*network*>	(Optional) Network interface (in local machine) where messages are received.
-daemon <*daemon*>	(Optional) Internal port of application to the rvd daemon (default: 7500).
<*subject_list*>	Subjects listen to.

To use the cns-listen command, follow these steps:

Step 1 Log in to the host system as root.

Step 2 Change directories to **/opt/CSCOcnsie/tools**.

Step 3 Type **./cns-listen** <*subject_list*>

Usage Guidelines

Use the greater than symbol (>) for a wildcard.

Examples

./cns-listen "cisco.cns.config.load"

./cns-listen "cisco.cns.>"

Software Licenses and Acknowledgements

For information on the third-party software licenses, see the *Cisco Configuration Engine 3.5.4 Open Source Documentation*.

Symbols

Numerics

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W

X