CHAPTER 5

Configuring Cisco IOS Configuration Engine

This chapter describes how to configure the feature on the switch.

Note For complete configuration information for the Cisco Configuration Engine, go to http://www.cisco.com/en/US/products/sw/netmgtsw/ps4617/tsd_products_support_series_home.html

For complete syntax and usage information for the commands used in this chapter, go to the *Cisco IOS Network Management Command Reference, Release 12.4* http://www.cisco.com/en/US/docs/ios/netmgmt/command/reference/nm_book.html

This chapter consists of these sections:

Understanding Cisco Configuration Engine Software

The Cisco Configuration Engine is network management software that acts as a configuration service for automating the deployment and management of network devices and services (see Figure 5-1). Each Configuration Engine manages a group of Cisco devices (switches and routers) and the services that they deliver, storing their configurations and delivering them as needed. The Configuration Engine automates initial configurations and configuration updates by generating device-specific configuration changes, sending them to the device, executing the configuration change, and logging the results.

The Configuration Engine supports standalone and server modes and has these CNS components:

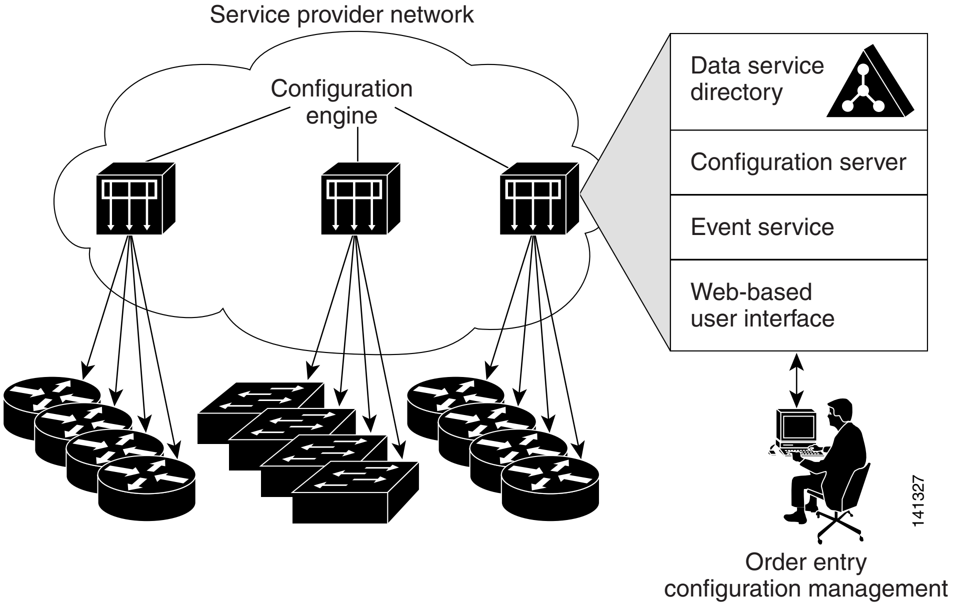
Configuration service (web server, file manager, and namespace mapping server)

Event service (event gateway)

Data service directory (data models and schema)

In standalone mode, the Configuration Engine supports an embedded Directory Service. In this mode, no external directory or other data store is required. In server mode, the Configuration Engine supports the use of a user-defined external directory.

Figure 5-1 Configuration Engine Architectural Overview



These sections contain this conceptual information:

- Configuration Service, page 5-2
- Event Service, page 5-3
- What You Should Know About the CNS IDs and Device Hostnames, page 5-3

Configuration Service

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

The Cisco IOS agent can perform a syntax check on received configuration files and publish events to show the success or failure of the syntax check. The configuration agent can either apply configurations immediately or delay the application until receipt of a synchronization event from the configuration server.

Event Service

uses subject-based addressing to send messages to their destinations. Subject-based addressing conventions define a simple, uniform namespace for messages and their destinations.

NameSpace Mapper

The Configuration Engine includes the NameSpace Mapper (NSM) that provides a lookup service for managing logical groups of devices based on application, device or group ID, and event.

Cisco IOS devices recognize only event subject-names that match those configured in Cisco IOS software; for example, cisco.cns.config.load. You can use the namespace mapping service to designate events by using any desired naming convention. When you have populated your data store with your subject names, NSM changes your event subject-name strings to those known by Cisco IOS.

For a subscriber, when given a unique device ID and event, the namespace mapping service returns a set of events to which to subscribe. Similarly, for a publisher, when given a unique group ID, device ID, and event, the mapping service returns a set of events on which to publish.

What You Should Know About the CNS IDs and Device Hostnames

ConfigID
DeviceID

Because the Configuration Engine uses both the event bus and the configuration server to provide configurations to devices, you must define both ConfigID and Device ID for each configured switch.

Within the scope of a single instance of the configuration server, no two configured switches can share the same value for ConfigID. Within the scope of a single instance of the event bus, no two configured switches can share the same value for DeviceID.

ConfigID

DeviceID

cns config partial

Hostname and DeviceID

no cns event

cns event

Caution

–not *before*

Using Hostname, DeviceID, and ConfigID

is sent on the cn=<*value*> of the device.

In server mode, the hostname is not used. In this mode, the unique DeviceID attribute is always used for sending an event on the bus. If this attribute is not set, you cannot update the switch.

These and other associated attributes (tag value pairs) are set when you run **Setup**

Understanding Cisco IOS Agents

•

•

•

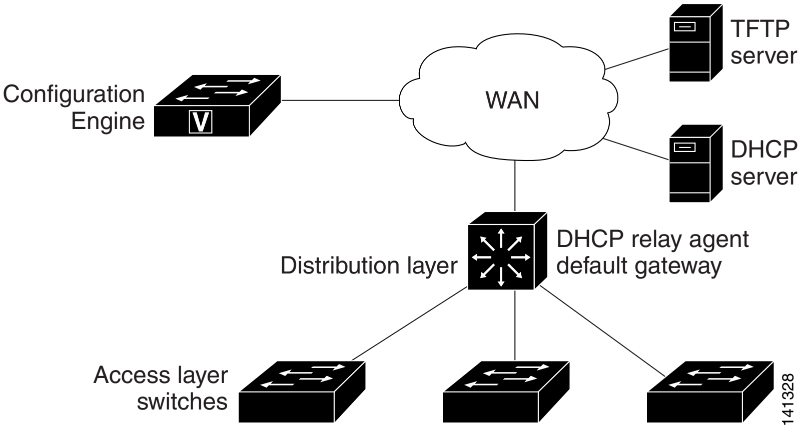
Initial Configuration

The switch automatically configures the assigned IP address on interface VLAN 1 (the default) and downloads the bootstrap configuration file from the TFTP server. Upon successful download of the bootstrap configuration file, the switch loads the file in its running configuration.

The Cisco IOS agents initiate communication with the Configuration Engine by using the appropriate ConfigID and EventID. The Configuration Engine maps the Config ID to a template and downloads the full configuration file to the switch.

Figure 5-2 shows a sample network configuration for retrieving the initial bootstrap configuration file by using DHCP-based autoconfiguration.

Figure 5-2 Initial Configuration Overview



Incremental (Partial) Configuration

Synchronized Configuration

Configuring Cisco IOS Agents

“Enabling Automated CNS Configuration” section on page 5-6. If you want to change the configuration or install a custom configuration, see these sections for instructions:

- Enabling the CNS Event Agent, page 5-8
- Enabling the Cisco IOS CNS Agent, page 5-9

To enable automated CNS configuration of the switch, you must first complete the prerequisites in Table 5-1. When you complete them, power on the switch. At the prompt, do nothing: The switch begins the initial configuration as described in the “Initial Configuration” section on page 5-5. When the full configuration file is loaded on your switch, you need to do nothing else.

Table 5-1 Prerequisites for Enabling Automatic Configuration

Device	Required Configuration
	• • •
	• • • •
	• • •

For more information about running the setup program and creating templates on the Configuration Engine, see the *Cisco Configuration Engine Installation and Setup Guide, 1.5 for Linux*

Enabling the CNS Event Agent

Note

Beginning in privileged EXEC mode, follow these steps to enable the CNS event agent on the switch:

	Command	Purpose
Step 1	**configure terminal**	
Step 2	} [ ] [**backup** **failover-time** **keepalive** *retry-count* **reconnect** *time* **source** *ip-address*	*hostname* *ip-address* *port number* **backup** **failover-time** *seconds* **keepalive** *seconds* *retry-count* **reconnect** *time* **source** *ip-address* **encrypt** **clock-timeout**
Step 3	**end**	
Step 4	**show cns event connections**	
Step 5	**show running-config**	
Step 6		

To disable the CNS event agent, use the {*ip-address* | *hostname*} global configuration command.

This example shows how to enable the CNS event agent, set the IP address gateway to 10.180.1.27, set 120 seconds as the keepalive interval, and set 10 as the retry count.

```
Switch(config)# cns event 10.180.1.27 keepalive 120 10
```

Enabling the Cisco IOS CNS Agent

-
-

Enabling an Initial Configuration

	Command	Purpose
Step 1		
Step 2		
Step 3		
Step 4		
Step 5	**exit**	
Step 6	**cns connect** **retries** **retry-interval** **sleep** **timeout**	• • **retries** • **retry-interva** • **sleep** • **timeout**

	Command	Purpose
Step 7		• • • •
Step 8		
Step 9		
Step 10		
Step 11		
Step 12		

	Command	Purpose
Step 13		• • • • **Note** • **hostname**| **string** **udi** **hardware-serial** **hostname** **string** **udi**

no cns config initial

configuration is unknown (the CNS Zero Touch feature).

```
                cns template connect template-dhcp
Switch(config-tmpl-conn)# cli ip address dhcp
Switch(config-tmpl-conn)# exit
                cns template connect ip-route
                          cli ip route 0.0.0.0 0.0.0.0 ${next-hop}
                          exit
                cns connect dhcp
Switch(config-cns-conn)# discover interface gigabitethernet
                         template template-dhcp
                         template ip-route
                         exit
                hostname RemoteSwitch
RemoteSwitch(config)#
```

```
Switch(config)#
Switch(config-tmpl-conn)#
Switch(config-tmpl-conn)#
Switch(config)#
Switch(config-tmpl-conn)#
Switch(config-tmpl-conn)#
Switch(config)#
Switch(config-cns-conn)#
Switch(config-cns-conn)#
Switch(config-cns-conn)#
Switch(config-cns-conn)#
Switch(config)#
RemoteSwitch(config)# ip route 172.28.129.22 255.255.255.255 11.11.11.1
                      cns id ethernet 0 ipaddress
                      cns config initial 172.28.129.22 no-persist
```

Enabling a Partial Configuration

	Command	Purpose
Step 1		
Step 2		• • • **Note**
Step 3		
Step 4		
Step 5		
Step 6		

Displaying CNS Configuration

Displaying CNS Configuration

show cns event subject	