



Configuring Cisco IOS Configuration Engine

This chapter describes how to configure the Cisco IOS Configuration Engine on the Cisco Industrial Ethernet 2000U Series (IE 2000U) and Connected Grid Switches, hereafter referred to as switch.



Note

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

For complete syntax and usage information for the commands used in this chapter, see the documents listed in the “[Related Documents](#)” section on [page 3-17](#).

- [Information About Cisco Configuration Engine Software, page 3-1](#)
- [Information About Cisco IOS Agents, page 3-5](#)
- [Prerequisites, page 3-6](#)
- [Guidelines and Limitations, page 3-7](#)
- [Default Settings, page 3-7](#)
- [Configuring Cisco IOS Agents, page 3-7](#)
- [Verifying Configuration, page 3-15](#)
- [Configuration Example, page 3-16](#)
- [Related Documents, page 3-17](#)
- [Feature History, page 3-17](#)

Information About 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 3-1](#)). 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.

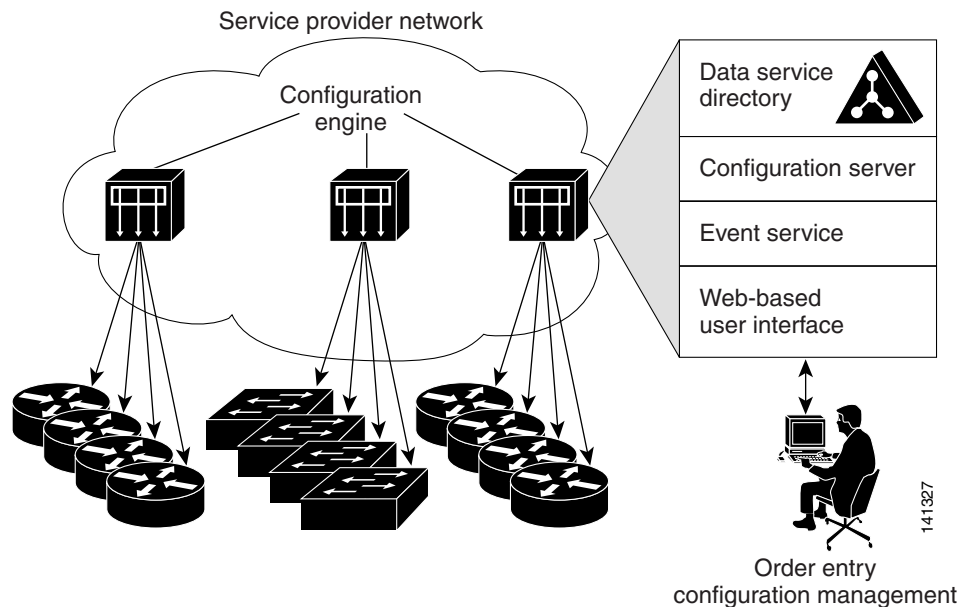
The Configuration Engine supports standalone and server modes and has these Cisco Networking Services (CNS) components:

- 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 3-1 Configuration Engine Architectural Overview



This section includes the following topics:

- [Configuration Service, page 3-2](#)
- [Event Service, page 3-3](#)
- [What You Should Know About the CNS IDs and Device Hostnames, page 3-3](#)

Configuration Service

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

The Configuration Service uses the CNS Event Service to send and receive configuration change events and to send success and failure notifications.

The configuration server is a web server that uses configuration templates and the device-specific configuration information stored in the embedded (standalone mode) or remote (server mode) directory.

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

The Cisco Configuration Engine uses the Event Service for receipt and generation of configuration events. The event agent is on the switch and facilitates the communication between the switch and the event gateway on the Configuration Engine.

The Event Service is a highly capable publish-and-subscribe communication method. The 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

The Configuration Engine assumes that a unique identifier is associated with each configured switch. This unique identifier can take on multiple synonyms, where each synonym is unique within a particular namespace. The event service uses namespace content for subject-based addressing of messages.

The Configuration Engine intersects two namespaces, one for the event bus and the other for the configuration server. Within the scope of the configuration server namespace, the term *ConfigID* is the unique identifier for a device. Within the scope of the event bus namespace, the term *DeviceID* is the CNS unique identifier for a device.

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

Each configured switch has a unique ConfigID, which serves as the key into the Configuration Engine directory for the corresponding set of switch CLI attributes. The ConfigID defined on the switch must match the ConfigID for the corresponding switch definition on the Configuration Engine.

The ConfigID is fixed at startup time and cannot be changed until the device restarts, even if the switch hostname is reconfigured.

DeviceID

Each configured switch participating on the event bus has a unique DeviceID, which is analogous to the switch source address so that the switch can be targeted as a specific destination on the bus. All switches configured with the **cns config partial** global configuration command must access the event bus. Therefore, the DeviceID, as originated on the switch, must match the DeviceID of the corresponding switch definition in the Configuration Engine.

The origin of the DeviceID is defined by the Cisco IOS hostname of the switch. However, the DeviceID variable and its usage reside within the event gateway adjacent to the switch.

The logical Cisco IOS termination point on the event bus is embedded in the event gateway, which in turn functions as a proxy on behalf of the switch. The event gateway represents the switch and its corresponding DeviceID to the event bus.

The switch declares its hostname to the event gateway immediately after the successful connection to the event gateway. The event gateway couples the DeviceID value to the Cisco IOS hostname each time this connection is established. The event gateway caches this DeviceID value for the duration of its connection to the switch.

Hostname and DeviceID

The DeviceID is fixed at the time of the connection to the event gateway and does not change even when the switch hostname is reconfigured.

When changing the switch hostname on the switch, the only way to refresh the DeviceID is to break the connection between the switch and the event gateway. Enter the **no cns event** global configuration command followed by the **cns event** global configuration command.

When the connection is re-established, the switch sends its modified hostname to the event gateway. The event gateway redefines the DeviceID to the new value.



Caution

When using the Configuration Engine user interface, you must first set the DeviceID field to the hostname value that the switch acquires *after*—not *before*—you use the **cns config initial** global configuration command at the switch. Otherwise, subsequent **cns config partial** global configuration command operations malfunction.

Using Hostname, DeviceID, and ConfigID

In standalone mode, when a hostname value is set for a switch, the configuration server uses the hostname as the DeviceID when an event is sent on hostname. If the hostname has not been set, the event 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** on the Configuration Engine.

For more information about running the setup program on the Configuration Engine, see the [Cisco Configuration Engine Installation and Configuration Guide](#).

Information About Cisco IOS Agents

The CNS event agent feature allows the switch to publish and subscribe to events on the event bus and works with the Cisco IOS agent. The Cisco IOS agent feature supports the switch by providing these features:

- [Initial Configuration, page 3-5](#)
- [Incremental \(Partial\) Configuration, page 3-6](#)
- [Synchronized Configuration, page 3-6](#)

Initial Configuration

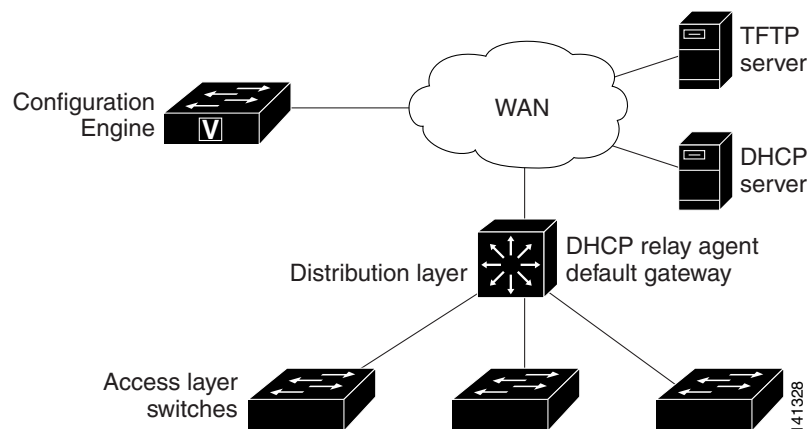
When the switch first comes up, it attempts to get an IP address by broadcasting a DHCP request on the network. Assuming there is no DHCP server on the subnet, the distribution switch acts as a DHCP relay agent and forwards the request to the DHCP server. Upon receiving the request, the DHCP server assigns an IP address to the new switch and includes the TFTP server IP address, the path to the bootstrap configuration file, and the default gateway IP address in a unicast reply to the DHCP relay agent. The DHCP relay agent forwards the reply to the switch.

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 3-2](#) shows a sample network configuration for retrieving the initial bootstrap configuration file by using DHCP-based autoconfiguration.

Figure 3-2 Initial Configuration Overview



Incremental (Partial) Configuration

After the network is running, new services can be added by using the Cisco IOS agent. Incremental (partial) configurations can be sent to the switch. The actual configuration can be sent as an event payload by way of the event gateway (push operation) or as a signal event that triggers the switch to initiate a pull operation.

The switch can check the syntax of the configuration before applying it. If the syntax is correct, the switch applies the incremental configuration and publishes an event that signals success to the configuration server. If the switch does not apply the incremental configuration, it publishes an event showing an error status. When the switch has applied the incremental configuration, it can write it to NVRAM or wait until signaled to do so.

Synchronized Configuration

When the switch receives a configuration, it can defer application of the configuration upon receipt of a write-signal event. The write-signal event tells the switch not to save the updated configuration into its NVRAM. The switch uses the updated configuration as its running configuration. This ensures that the switch configuration is synchronized with other network activities before saving the configuration in NVRAM for use at the next reboot.

Prerequisites

To enable automated CNS configuration of the switch, you must first complete the prerequisites in [Table 3-1](#). When you complete them, power on the switch. At the **setup** prompt, do nothing: The switch begins the initial configuration as described in the “[Initial Configuration](#)” section on page 3-5. When the full configuration file is loaded on your switch, you do not need to do anything else.

Table 3-1 Prerequisites for Enabling Automatic Configuration

Device	Required Configuration
Access switch	Factory default (no configuration file)
Distribution switch	<ul style="list-style-type: none"> • IP helper address • Enable DHCP relay agent • IP routing (if used as default gateway)
DHCP server	<ul style="list-style-type: none"> • IP address assignment • TFTP server IP address • Path to bootstrap configuration file on the TFTP server • Default gateway IP address

Table 3-1 Prerequisites for Enabling Automatic Configuration (continued)

Device	Required Configuration
TFTP server	<ul style="list-style-type: none"> • A bootstrap configuration file that includes the CNS configuration commands that enable the switch to communicate with the Configuration Engine • The switch configured to use either the switch MAC address or the serial number (instead of the default hostname) to generate the ConfigID and EventID • The CNS event agent configured to push the configuration file to the switch
CNS Configuration Engine	One or more templates for each type of device, with the ConfigID of the device mapped to the template.

**Note**

For more information about running the setup program and creating templates on the Configuration Engine, see the [Cisco Configuration Engine Installation and Configuration Guide](#).

Guidelines and Limitations

These restrictions apply to the image agent running on the switch:

- You can only download the tar image file. Downloading the bin image file is not supported.
- Only the immediate download option is supported. You cannot schedule a download to occur at a specified date and time.
- The Destination field in the Associate Image with Device window of the Cisco Configuration Engine GUI is not supported.

Default Settings

- CNS Event Agent: Disabled
- CNS Configuration Agent: Disabled

Configuring Cisco IOS Agents

The Cisco IOS agents embedded in the switch Cisco IOS software allow the switch to be connected and automatically configured as described in the “Prerequisites” section on page 3-6. If you want to change the configuration or install a custom configuration, see these sections for instructions:

- [Enabling the CNS Event Agent, page 3-8](#)
- [Enabling the Cisco IOS CNS Agent, page 3-9](#)
- [Upgrading Devices with Cisco IOS Image Agent, page 3-14](#)

Enabling the CNS Event Agent

You must enable the CNS event agent on the switch before you enable the CNS configuration agent.

BEFORE YOU BEGIN

Review the [“Information About Cisco IOS Agents”](#) section on page 3-5.

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal	Enter global configuration mode.
Step 2	cns event { <i>hostname ip-address</i> } [<i>port-number</i>] [backup] [failover-time <i>seconds</i>] [keepalive <i>seconds</i> <i>retry-count</i>] [reconnect <i>time</i>] [source <i>ip-address</i>]	<p>Enable the event agent, and enter the gateway parameters.</p> <ul style="list-style-type: none"> For {<i>hostname ip-address</i>}, enter either the hostname or the IP address of the event gateway. (Optional) For <i>port number</i>, enter the port number for the event gateway. The default port number is 11011. (Optional) Enter backup to show that this is the backup gateway. (If omitted, this is the primary gateway.) (Optional) For failover-time <i>seconds</i>, enter how long the switch waits for the primary gateway route after the route to the backup gateway is established. (Optional) For keepalive <i>seconds</i>, enter how often the switch sends keepalive messages. For <i>retry-count</i>, enter the number of unanswered keepalive messages that the switch sends before the connection is terminated. The default for each is 0. (Optional) For reconnect <i>time</i>, enter the maximum time interval that the switch waits before trying to reconnect to the event gateway. (Optional) For source <i>ip-address</i>, enter the source IP address of this device. <p>Note Though visible in the command-line help string, the encrypt and the clock-timeout <i>time</i> keywords are not supported.</p>
Step 3	end	Return to privileged EXEC mode.
Step 4	show cns event connections	Verify information about the event agent.

	Command	Purpose
Step 5	show running-config	Verify your entries.
Step 6	copy running-config startup-config	(Optional) Save your entries in the configuration file.

To disable the CNS event agent, use the **no cns event** {*ip-address* | *hostname*} global configuration command.

EXAMPLE

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

After enabling the CNS event agent, start the Cisco IOS CNS agent on the switch. You can enable the Cisco IOS agent with these commands:

- The **cns config initial** global configuration command enables the Cisco IOS agent and initiates an initial configuration on the switch.
- The **cns config partial** global configuration command enables the Cisco IOS agent and initiates a partial configuration on the switch. You can then use the Configuration Engine to remotely send incremental configurations to the switch.

Enabling an Initial Configuration

BEFORE YOU BEGIN

Complete the [“Enabling the CNS Event Agent” procedure on page 3-8](#).

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal	Enter global configuration mode.
Step 2	cns template connect <i>name</i>	Enter CNS template connect configuration mode, and specify the name of the CNS connect template.
Step 3	cli <i>config-text</i>	Enter a command line for the CNS connect template. Repeat this step for each command line in the template.
Step 4		Repeat Steps 2 to 3 to configure another CNS connect template.
Step 5	exit	Return to global configuration mode.

	Command	Purpose
Step 6	cns connect <i>name</i> [retries <i>number</i>] [retry-interval <i>seconds</i>] [sleep <i>seconds</i>] [timeout <i>seconds</i>]	Enter CNS connect configuration mode, specify the name of the CNS connect profile, and define the profile parameters. The switch uses the CNS connect profile to connect to the Configuration Engine. <ul style="list-style-type: none"> • Enter the name of the CNS connect profile. • (Optional) For retries <i>number</i>, enter the number of connection retries. The range is 1 to 30. The default is 3. • (Optional) For retry-interval <i>seconds</i>, enter the interval between successive connection attempts to the Configuration Engine. The range is 1 to 40 seconds. The default is 10 seconds. • (Optional) For sleep <i>seconds</i>, enter the amount of time before which the first connection attempt occurs. The range is 0 to 250 seconds. The default is 0. • (Optional) For timeout <i>seconds</i>, enter the amount of time after which the connection attempts end. The range is 10 to 2000 seconds. The default is 120.
Step 7	discover { controller <i>controller-type</i> dcli [subinterface <i>subinterface-number</i>] interface [<i>interface-type</i>] line <i>line-type</i> }	Specify the interface parameters in the CNS connect profile. <ul style="list-style-type: none"> • For controller <i>controller-type</i>, enter the controller type. • For dcli, enter the active data-link connection identifiers (DLCIs). (Optional) For subinterface <i>subinterface-number</i>, specify the point-to-point subinterface number that is used to search for active DLCIs. • For interface [<i>interface-type</i>], enter the type of interface. • For line <i>line-type</i>, enter the line type.
Step 8	template <i>name</i> [... <i>name</i>]	Specify the list of CNS connect templates in the CNS connect profile to be applied to the switch configuration. You can specify more than one template.
Step 9		Repeat Steps 7 to 8 to specify more interface parameters and CNS connect templates in the CNS connect profile.
Step 10	exit	Return to global configuration mode.
Step 11	hostname <i>name</i>	Enter the hostname for the switch.
Step 12	ip route <i>network-number</i>	(Optional) Establish a static route to the Configuration Engine whose IP address is <i>network-number</i> .

Command	Purpose
<p>Step 13 <code>cns id interface num { dns-reverse ipaddress mac-address } [event] [image]</code></p> <p>or</p> <p><code>cns id { hardware-serial hostname string string udi } [event] [image]</code></p>	<p>(Optional) Set the unique EventID or ConfigID used by the Configuration Engine.</p> <ul style="list-style-type: none"> For <i>interface num</i>, enter the type of interface—for example, ethernet, group-async, loopback, or virtual-template. This setting specifies from which interface the IP or MAC address should be retrieved to define the unique ID. For { dns-reverse ipaddress mac-address }, enter dns-reverse to retrieve the hostname and assign it as the unique ID, enter ipaddress to use the IP address, or enter mac-address to use the MAC address as the unique ID. (Optional) Enter event to set the ID to be the event-id value used to identify the switch. (Optional) Enter image to set the ID to be the image-id value used to identify the switch. <p>Note If both the event and image keywords are omitted, the image-id value is used to identify the switch.</p> <ul style="list-style-type: none"> For { hardware-serial hostname string string udi }, enter hardware-serial to set the switch serial number as the unique ID, enter hostname (the default) to select the switch hostname as the unique ID, enter an arbitrary text string for string string as the unique ID, or enter udi to set the unique device identifier (UDI) as the unique ID.

	Command	Purpose
Step 14	cns config initial {hostname ip-address} [port-number] [event] [no-persist] [page page] [source ip-address] [syntax-check]	<p>Enable the Cisco IOS agent, and initiate an initial configuration.</p> <ul style="list-style-type: none"> For {hostname ip-address}, enter the hostname or the IP address of the configuration server. (Optional) For <i>port-number</i>, enter the port number of the configuration server. The default port number is 80. (Optional) Enable event for configuration success, failure, or warning messages when the configuration is finished. (Optional) Enable no-persist to suppress the automatic writing to NVRAM of the configuration pulled as a result of entering the cns config initial global configuration command. If the no-persist keyword is not entered, using the cns config initial command causes the resultant configuration to be automatically written to NVRAM. (Optional) For <i>page page</i>, enter the web page of the initial configuration. The default is /Config/config/asp. (Optional) Enter source ip-address to use for source IP address. (Optional) Enable syntax-check to check the syntax when this parameter is entered. <p>Note Though visible in the command-line help string, the encrypt, status url, and inventory keywords are not supported.</p>
Step 15	end	Return to privileged EXEC mode.
Step 16	show cns config connections	Verify information about the configuration agent.
Step 17	show running-config	Verify your entries.

To disable the CNS Cisco IOS agent, use the **no cns config initial** {ip-address | hostname} global configuration command.

EXAMPLE

This example shows how to configure an initial configuration on a remote switch when the switch configuration is unknown (the CNS Zero Touch feature):

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

```
Switch(config-cns-conn)# discover interface gigabitethernet
Switch(config-cns-conn)# template template-dhcp
Switch(config-cns-conn)# template ip-route
Switch(config-cns-conn)# exit
Switch(config)# hostname RemoteSwitch
RemoteSwitch(config)# cns config initial 10.1.1.1 no-persist
```

This example shows how to configure an initial configuration on a remote switch when the switch IP address is known. The Configuration Engine IP address is 172.28.129.22.

```
Switch(config)# cns template connect template-dhcp
Switch(config-tmpl-conn)# cli ip address dhcp
Switch(config-tmpl-conn)# exit
Switch(config)# cns template connect ip-route
Switch(config-tmpl-conn)# cli ip route 0.0.0.0 0.0.0.0 ${next-hop}
Switch(config-tmpl-conn)# exit
Switch(config)# cns connect dhcp
Switch(config-cns-conn)# discover interface gigabitethernet
Switch(config-cns-conn)# template template-dhcp
Switch(config-cns-conn)# template ip-route
Switch(config-cns-conn)# exit
Switch(config)# hostname RemoteSwitch
RemoteSwitch(config)# ip route 172.28.129.22 255.255.255.255 11.11.11.1
RemoteSwitch(config)# cns id ethernet 0 ipaddress
RemoteSwitch(config)# cns config initial 172.28.129.22 no-persist
```

Enabling a Partial Configuration

BEFORE YOU BEGIN

Complete the “[Enabling the CNS Event Agent](#)” procedure on page 3-8.

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal	Enter global configuration mode.
Step 2	cns config partial { <i>ip-address</i> <i>hostname</i> } [<i>port-number</i>] [<i>source ip-address</i>]	<p>Enable the configuration agent, and initiate a partial configuration.</p> <ul style="list-style-type: none"> For {<i>ip-address</i> <i>hostname</i>}, enter the IP address or the hostname of the configuration server. (Optional) For <i>port-number</i>, enter the port number of the configuration server. The default port number is 80. (Optional) Enter source ip-address to use for the source IP address. <p>Note Though visible in the command-line help string, the encrypt keyword is not supported.</p>
Step 3	end	Return to privileged EXEC mode.

	Command	Purpose
Step 4	show cns config stats or show cns config outstanding	Verify information about the configuration agent.
Step 5	show running-config	Verify your entries.
Step 6	copy running-config startup-config	(Optional) Save your entries in the configuration file.

To disable the Cisco IOS agent, use the **no cns config partial** *{ip-address | hostname}* global configuration command. To cancel a partial configuration, use the **cns config cancel** privileged EXEC command.

EXAMPLE

The following example shows how to configure the CNS partial configuration agent to accept events from the event gateway at 172.28.129.22. The CNS partial configuration agent will connect to the CNS configuration server at 172.28.129.22, port number 80. The CNS partial configuration agent requests are redirected to a configuration server at 172.28.129.40, port number 80.

```
Switch(config)# cns event 172.28.129.22
Switch(config)# cns trusted-server config 172.28.129.40
Switch(config)# cns config partial 172.28.129.22
```

Upgrading Devices with Cisco IOS Image Agent

Administrators maintaining large networks of Cisco IOS devices need an automated mechanism to load image files onto large numbers of remote devices. Existing network management applications are useful to determine which images to run and how to manage images received from the Cisco online software center. Other image distribution solutions do not scale to cover thousands of devices and cannot distribute images to devices behind a firewall. The CNS image agent enables the managed device to initiate a network connection and request an image download allowing devices behind firewalls to access the image server.

You can use image agent to download one or more devices. The switches must have the image agent running on them.

BEFORE YOU BEGIN

Confirm these prerequisites before upgrading one or more devices with image agent:

- Determine where to store the Cisco IOS images on a file server to make the image available to the other networking devices. If the CNS Event Bus is to be used to store and distribute the images, the CNS event agent must be configured.
- Set up a file server to enable the networking devices to download the new images using the HTTPS protocol.
- Determine how to handle error messages generated by image agent operations. Error messages can be sent to the CNS Event Bus or an HTTP or HTTPS URL.

During automated image loading operations you must try to prevent the Cisco IOS device from losing connectivity with the file server that is providing the image. Image reloading is subject to memory issues and connection issues. Boot options must also be configured to allow the Cisco IOS device to boot another image if the first image reload fails.

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal	Enter global configuration mode
Step 2	ip host {ip-address} {hostname}	Enter the IP address and the hostname of the event gateway.
Step 3	cns trusted-server all-agents {hostname}	Specify a trusted server for CNS agent.
Step 4	no cns aaa enable cns event {ip-address} {port number}	Disable AAA authentication on the event gateway.
Step 5	cns image retry {number}	Specify the number of times to retry and download the image.
Step 6	cns image server {ip-address} status {ip-address}	Download the image from the server to the switch.
Step 7	end	Return to privileged EXEC mode.

EXAMPLE

This example shows how to upgrade a switch from a server with the address of **172.20.249.20**:

```
Switch(config)> configure terminal
Switch(config)# ip host cns-dsbu.cisco.com 172.20.249.20
Switch(config)# cns trusted-server all-agents cns-dsbu.cisco.com
Switch(config)# no cns aaa enable cns event 172.20.249.20 22022
Switch(config)# cns image retry 1
Switch(config)# cns image server http://172.20.249.20:80/cns/HttpMsgDispatcher status
http://172.20.249.20:80/cns/HttpMsgDispatcher
Switch(config)# end
```

Verifying Configuration

Command	Purpose
show cns config connections	Displays the status of the CNS Cisco IOS agent connections.
show cns config outstanding	Displays information about incremental (partial) CNS configurations that have started but are not yet completed.
show cns config stats	Displays statistics about the Cisco IOS agent.
show cns event connections	Displays the status of the CNS event agent connections.
show cns event stats	Displays statistics about the CNS event agent.
show cns event subject	Displays a list of event agent subjects that are subscribed to by applications.
show cns image	Displays the status of the image download.

Configuration Example

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
```

This example shows how to configure an initial configuration on a remote switch when the switch configuration is unknown (the CNS Zero Touch feature):

```
Switch(config)# cns template connect template-dhcp
Switch(config-tmpl-conn)# cli ip address dhcp
Switch(config-tmpl-conn)# exit
Switch(config)# cns template connect ip-route
Switch(config-tmpl-conn)# cli ip route 0.0.0.0 0.0.0.0 ${next-hop}
Switch(config-tmpl-conn)# exit
Switch(config)# cns connect dhcp
Switch(config-cns-conn)# discover interface gigabitethernet
Switch(config-cns-conn)# template template-dhcp
Switch(config-cns-conn)# template ip-route
Switch(config-cns-conn)# exit
Switch(config)# hostname RemoteSwitch
RemoteSwitch(config)# cns config initial 10.1.1.1 no-persist
```

This example shows how to configure an initial configuration on a remote switch when the switch IP address is known. The Configuration Engine IP address is 172.28.129.22.

```
Switch(config)# cns template connect template-dhcp
Switch(config-tmpl-conn)# cli ip address dhcp
Switch(config-tmpl-conn)# exit
Switch(config)# cns template connect ip-route
Switch(config-tmpl-conn)# cli ip route 0.0.0.0 0.0.0.0 ${next-hop}
Switch(config-tmpl-conn)# exit
Switch(config)# cns connect dhcp
Switch(config-cns-conn)# discover interface gigabitethernet
Switch(config-cns-conn)# template template-dhcp
Switch(config-cns-conn)# template ip-route
Switch(config-cns-conn)# exit
Switch(config)# hostname RemoteSwitch
RemoteSwitch(config)# ip route 172.28.129.22 255.255.255.255 11.11.11.1
RemoteSwitch(config)# cns id ethernet 0 ipaddress
RemoteSwitch(config)# cns config initial 172.28.129.22 no-persist
```

The following example shows how to configure the CNS partial configuration agent to accept events from the event gateway at 172.28.129.22. The CNS partial configuration agent will connect to the CNS configuration server at 172.28.129.22, port number 80. The CNS partial configuration agent requests are redirected to a configuration server at 172.28.129.40, port number 80.

```
Switch(config)# cns event 172.28.129.22
Switch(config)# cns trusted-server config 172.28.129.40
Switch(config)# cns config partial 172.28.129.22
```

This example shows how to upgrade a switch from a server with the address of 172.20.249.20:

```
Switch(config)> configure terminal
Switch(config)# ip host cns-dsbu.cisco.com 172.20.249.20
Switch(config)# cns trusted-server all-agents cns-dsbu.cisco.com
Switch(config)# no cns aaa enable cns event 172.20.249.20 22022
Switch(config)# cns image retry 1
Switch(config)# cns image server http://172.20.249.20:80/cns/HttpMsgDispatcher status
http://172.20.249.20:80/cns/HttpMsgDispatcher
Switch(config)# end
```


Related Documents

- [Cisco IOS Master Command List, All Releases](#)
- [Cisco IOS Cisco Networking Services Command Reference](#)
- [Cisco Configuration Engine Installation and Configuration Guide](#)
- [Cisco Configuration Engine Administration Guide](#)
- [Configuration Fundamentals Configuration Guide, Cisco IOS Release 15M&T](#)

Feature History

Platform	First Supported Release
IE 2000U	Cisco IOS Release 15.0(2)EH
CGS 2520	Cisco IOS Release 12.2(53)EX
Ethernet Switch Module (ESM) for CGR 2010	Cisco IOS Release 12.2(53)EX

