Configuring Cisco IOS Configuration Engine

- Finding Feature Information, on page 1
- Prerequisites for Configuring the Configuration Engine, on page 1
- Restrictions for Configuring the Configuration Engine, on page 2
- Information About Configuring the Configuration Engine, on page 2
- How to Configure the Configuration Engine, on page 7
- Monitoring CNS Configurations, on page 20

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Configuring the Configuration Engine

- Obtain the name of the configuration engine instance to which you are connecting.
- Because the CNS 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 device.
- All devices configured with the **cns config partial** global configuration command must access the event bus. The DeviceID, as originated on the device, must match the DeviceID of the corresponding device definition in the Cisco Configuration Engine. You must know the hostname of the event bus to which you are connecting.

Related Topics

Cisco Networking Services IDs and Device Hostnames, on page 4

DeviceID, on page 4

Restrictions for Configuring the Configuration Engine

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

Related Topics

Cisco Networking Services IDs and Device Hostnames, on page 4

Information About Configuring the Configuration Engine

Cisco Configuration Engine Software

The Cisco Configuration Engine is network management utility software that acts as a configuration service for automating the deployment and management of network devices and services. Each Cisco Configuration Engine manages a group of Cisco devices (devices and routers) and the services that 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, sending them to the device, executing the configuration change, and logging the results.

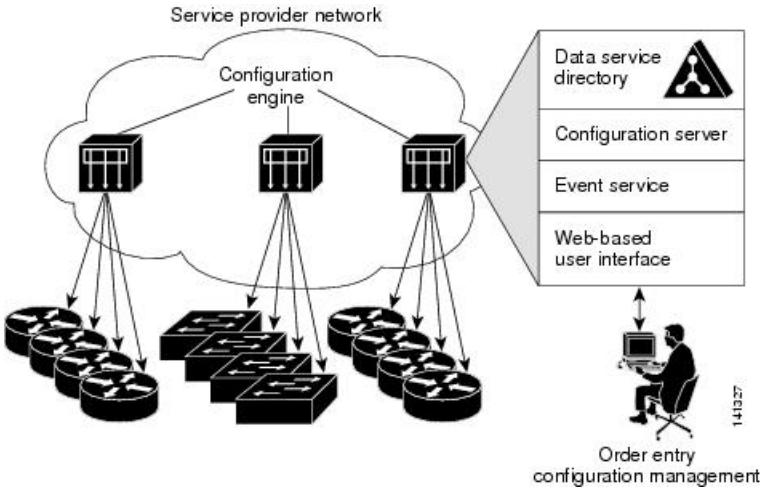
The Cisco Configuration Engine supports standalone and server modes and has these Cisco Networking Services (CNS) components:

- Configuration service:
  - Web server
  - File manager
  - Namespace mapping server
- Event service (event gateway)
- Data service directory (data models and schema)

Note Support for Cisco Configuration Engine will be deprecated in future releases. Use the configuration described in Cisco Plug and Play Feature Guide .

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

Figure 1: Cisco Configuration Engine Architectural Overview



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 device. The Configuration Service delivers device and service configurations to the device for initial configuration and mass reconfiguration by logical groups. Devices 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 by 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 Service consists of an event agent and an event gateway. The event agent is on the device and facilitates the communication between the device and the event gateway on the Cisco 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.

Related Topics

Enabling the CNS Event Agent, on page 7

NameSpace Mapper

The Cisco 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.

Cisco Networking Services IDs and Device Hostnames

The Cisco Configuration Engine assumes that a unique identifier is associated with each configured device. 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 Cisco 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.

Related Topics

Prerequisites for Configuring the Configuration Engine, on page 1
Restrictions for Configuring the Configuration Engine, on page 2

ConfigID

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

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

DeviceID

Each configured device participating on the event bus has a unique DeviceID, which is analogous to the device source address so that the device can be targeted as a specific destination on the bus.

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

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 device. The event gateway represents the device and its corresponding DeviceID to the event bus.

The device 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 retains this DeviceID value for the duration of its connection to the device.

Related Topics

Prerequisites for Configuring the Configuration Engine, on page 1

Hostname and DeviceID

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

When changing the device hostname on the device, the only way to refresh the DeviceID is to break the connection between the device and the event gateway. For instructions on refreshing DeviceIDs, see "Related Topics."

When the connection is reestablished, the device sends its modified hostname to the event gateway. The event gateway redefines the DeviceID to the new value.

Caution When using the Cisco Configuration Engine user interface, you must first set the DeviceID field to the hostname value that the device acquires *after*, not *before*, and you must reinitialize the configuration for your Cisco IOS CNS agent. Otherwise, subsequent partial configuration command operations may malfunction.

Related Topics

Refreshing DeviceIDs, on page 16

Hostname, DeviceID, and ConfigID

In standalone mode, when a hostname value is set for a device, 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 device.

These and other associated attributes (tag value pairs) are set when you run **Setup** on the Cisco Configuration Engine.

Cisco IOS CNS Agents

The CNS event agent feature allows the device to publish and subscribe to events on the event bus and works with the Cisco IOS CNS agent. These agents, embedded in the device Cisco IOS software, allow the device to be connected and automatically configured.

Related Topics

Enabling the Cisco IOS CNS Agent, on page 9

Initial Configuration

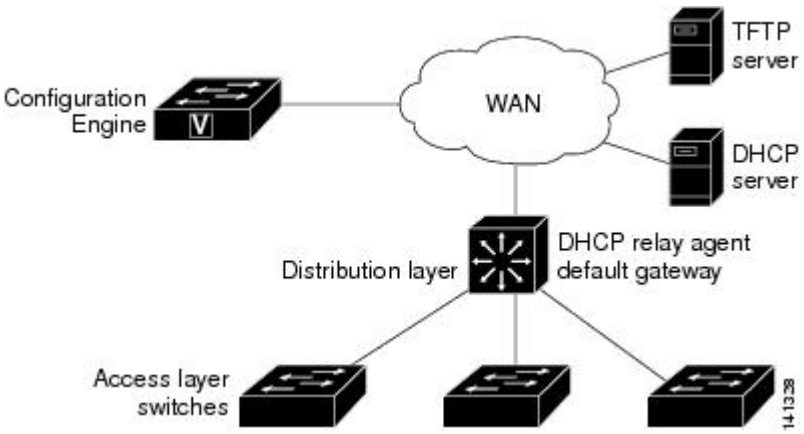
When the device first comes up, it attempts to get an IP address by broadcasting a Dynamic Host Configuration Protocol (DHCP) request on the network. Assuming there is no DHCP server on the subnet, the distribution device acts as a DHCP relay agent and forwards the request to the DHCP server. Upon receiving the request, the DHCP server assigns an IP address to the new device and includes the Trivial File Transfer Protocol (TFTP) server Internet Protocol (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 device.

The device 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 device loads the file in its running configuration.

The Cisco IOS CNS 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 device.

The following figure shows a sample network configuration for retrieving the initial bootstrap configuration file by using DHCP-based autoconfiguration.

Figure 2: Initial Configuration



Related Topics

Enabling an Initial Configuration for Cisco IOS CNS Agent, on page 11
Monitoring CNS Configurations, on page 20

Incremental (Partial) Configuration

After the network is running, new services can be added by using the Cisco IOS CNS agent. Incremental (partial) configurations can be sent to the device. 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 device to initiate a pull operation.

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

Related Topics

Enabling a Partial Configuration for Cisco IOS CNS Agent, on page 18
Monitoring CNS Configurations, on page 20

Synchronized Configuration

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

Automated CNS Configuration

To enable automated CNS configuration of the device, you must first complete the prerequisites listed in this topic. When you complete them, power on the device. At the **setup** prompt, do nothing; the device begins the initial configuration. When the full configuration file is loaded on your device, you do not need to do anything else.

For more information on what happens during initial configuration, see "Related Topics."

Table 1: Prerequisites for Enabling Automatic Configuration

Device	Required Configuration
Access device	Factory default (no configuration file)
Distribution device	• IP helper address • Enable DHCP relay agent[1] • IP routing (if used as default gateway)
DHCP server	• IP address assignment • TFTP server IP address • Path to bootstrap configuration file on the TFTP server • Default gateway IP address
TFTP server	• A bootstrap configuration file that includes the CNS configuration commands that enable the device to communicate with the Configuration Engine • The device configured to use either the device 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 device
CNS Configuration Engine	One or more templates for each type of device, with the ConfigID of the device mapped to the template.

[1] A DHCP Relay is needed only when the DHCP Server is on a different subnet from the client.

How to Configure the Configuration Engine

Enabling the CNS Event Agent

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

Follow these steps to enable the CNS event agent on the device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cns event** {*hostname* | *ip-address*} [*port-number*] [[**keepalive** *seconds retry-count*] [**failover-time** *seconds*] [**reconnect-time** *time*] | **backup**]
4. **end**
5. **show running-config**
6. **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**cns event** {*hostname* | *ip-address*} [*port-number*] [[**keepalive** *seconds retry-count*] [**failover-time** *seconds*] [**reconnect-time** *time*] | **backup**] **Example:** `Device(config)# cns event 10.180.1.27 keepalive 120 10`	Enables the event agent, and enters the gateway parameters. • For {*hostname* | *ip-address*}, enter either the hostname or the IP address of the event gateway. • (Optional) For *port number*, enter the port number for the event gateway. The default port number is 11011. • (Optional) For **keepalive** *seconds*, enter how often the device sends keepalive messages. For *retry-count*, enter the number of unanswered keepalive messages that the device sends before the connection is terminated. The default for each is 0. • (Optional) For **failover-time** *seconds*, enter how long the device waits for the primary gateway route after the route to the backup gateway is established. • (Optional) For **reconnect-time** *time*, enter the maximum time interval that the device waits before trying to reconnect to the event gateway.

	Command or Action	Purpose
		• (Optional) Enter **backup** to show that this is the backup gateway. (If omitted, this is the primary gateway.) **Note** Though visible in the command-line help string, the **encrypt** and the **clock-timeout** *time* keywords are not supported.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

What to do next

To verify information about the event agent, use the **show cns event connections** command in privileged EXEC mode.

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

Related Topics

Event Service, on page 3

Enabling the Cisco IOS CNS Agent

Follow these steps to enable the Cisco IOS CNS agent on the device.

Before you begin

You must enable the CNS event agent on the device before you enable this agent.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **cns config initial** {*hostname* | *ip-address*} [*port-number*]
4. **cns config partial** {*hostname* | *ip-address*} [*port-number*]
5. **end**
6. **show running-config**
7. **copy running-config startup-config**
8. Start the Cisco IOS CNS agent on the device.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**cns config initial** {*hostname* | *ip-address*} [*port-number*] **Example:** `Device(config)# cns config initial 10.180.1.27 10`	Enables the Cisco IOS CNS agent, and enters the configuration server parameters. • For {*hostname* | *ip-address*}, enter either the hostname or the IP address of the configuration server. • (Optional) For *port number*, enter the port number for the configuration server. This command enables the Cisco IOS CNS agent and initiates an initial configuration on the device.
Step 4	**cns config partial** {*hostname* | *ip-address*} [*port-number*] **Example:** `Device(config)# cns config partial 10.180.1.27 10`	Enables the Cisco IOS CNS agent, and enters the configuration server parameters. • For {*hostname* | *ip-address*}, enter either the hostname or the IP address of the configuration server. • (Optional) For *port number*, enter the port number for the configuration server. Enables the Cisco IOS CNS agent and initiates a partial configuration on the device.
Step 5	**end** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config)# end`	
Step 6	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.
Step 8	Start the Cisco IOS CNS agent on the device.	

What to do next

You can now use the Cisco Configuration Engine to remotely send incremental configurations to the device.

Related Topics

Cisco IOS CNS Agents, on page 5

Enabling an Initial Configuration for Cisco IOS CNS Agent

Follow these steps to enable the CNS configuration agent and initiate an initial configuration on the device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cns template connect** *name*
4. **cli** *config-text*
5. Repeat Steps 3 to 4 to configure another CNS connect template.
6. **exit**
7. **cns connect** *name* [**retries** *number*] [**retry-interval** *seconds*] [**sleep** *seconds*] [**timeout** *seconds*]
8. **discover** {**controller** *controller-type* | **dlci** [**subinterface** *subinterface-number*] | **interface** [*interface-type*] | **line** *line-type*}
9. **template** *name* [... *name*]
10. Repeat Steps 8 to 9 to specify more interface parameters and CNS connect templates in the CNS connect profile.
11. **exit**
12. **hostname** *name*
13. **ip route** *network-number*
14. **cns id** *interface num* {**dns-reverse** | **ipaddress** | **mac-address**} [**event**] [**image**]
15. **cns id** {**hardware-serial** | **hostname** | **string** *string* | **udi**} [**event**] [**image**]

16. **cns config initial** {*hostname* | *ip-address*} [*port-number*] [**event**] [**no-persist**] [**page** *page*] [**source** *ip-address*] [**syntax-check**]
17. **end**
18. **show running-config**
19. **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**cns template connect** *name* **Example:** `Device(config)# cns template connect template-dhcp`	Enters CNS template connect configuration mode, and specifies the name of the CNS connect template.
Step 4	**cli** *config-text* **Example:** `Device(config-tmpl-conn)# cli ip address dhcp`	Enters a command line for the CNS connect template. Repeat this step for each command line in the template.
Step 5	Repeat Steps 3 to 4 to configure another CNS connect template.	
Step 6	**exit** **Example:** `Device(config)# exit`	Returns to global configuration mode.
Step 7	**cns connect** *name* [**retries** *number*] [**retry-interval** *seconds*] [**sleep** *seconds*] [**timeout** *seconds*] **Example:** `Device(config)# cns connect dhcp`	Enters CNS connect configuration mode, specifies the name of the CNS connect profile, and defines the profile parameters. The device uses the CNS connect profile to connect to the Configuration Engine. • Enter the *name* of the CNS connect profile.

	Command or Action	Purpose
		• (Optional) For **retries** *number*, enter the number of connection retries. The range is 1 to 30. The default is 3. • (Optional) For **retry-interval** *seconds*, 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** *seconds*, 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** *seconds*, enter the amount of time after which the connection attempts end. The range is 10 to 2000 seconds. The default is 120.
Step 8	**discover** {**controller** *controller-type* | **dlci** [**subinterface** *subinterface-number*] | **interface** [*interface-type*] | **line** *line-type*} **Example:** `Device(config-cns-conn)# discover interface gigabitethernet`	Specifies the interface parameters in the CNS connect profile. • For **controller** *controller-type*, enter the controller type. • For **dlci**, enter the active data-link connection identifiers (DLCIs). (Optional) For **subinterface** *subinterface-number*, specify the point-to-point subinterface number that is used to search for active DLCIs. • For **interface** [*interface-type*], enter the type of interface. • For **line** *line-type*, enter the line type.
Step 9	**template** *name* [... *name*] **Example:** `Device(config-cns-conn)# template template-dhcp`	Specifies the list of CNS connect templates in the CNS connect profile to be applied to the device configuration. You can specify more than one template.
Step 10	Repeat Steps 8 to 9 to specify more interface parameters and CNS connect templates in the CNS connect profile.	
Step 11	**exit** **Example:** `Device(config-cns-conn)# exit`	Returns to global configuration mode.
Step 12	**hostname** *name* **Example:** `Device(config)# hostname device1`	Enters the hostname for the device.

	Command or Action	Purpose
Step 13	**ip route** *network-number* **Example:** `RemoteDevice(config)# ip route 172.28.129.22 255.255.255.255 11.11.11.1`	(Optional) Establishes a static route to the Configuration Engine whose IP address is *network-number*.
Step 14	**cns id** *interface num* {**dns-reverse** | **ipaddress** | **mac-address**} [**event**] [**image**] **Example:** `RemoteDevice(config)# cns id GigabitEthernet1/0/1 ipaddress`	(Optional) Sets the unique EventID or ConfigID used by the Configuration Engine. If you enter this command, do not enter the **cns id** {**hardware-serial** | **hostname** | **string** *string* | **udi**} [**event**] [**image**] command. • For *interface num*, 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 device. • (Optional) Enter **image** to set the ID to be the image-id value used to identify the device. **Note** If both the **event** and **image** keywords are omitted, the image-id value is used to identify the device.
Step 15	**cns id** {**hardware-serial** | **hostname** | **string** *string* | **udi**} [**event**] [**image**] **Example:** `RemoteDevice(config)# cns id hostname`	(Optional) Sets the unique EventID or ConfigID used by the Configuration Engine. If you enter this command, do not enter the **cns id** *interface num* {**dns-reverse** | **ipaddress** | **mac-address**} [**event**] [**image**] command. • For { **hardware-serial** | **hostname** | **string** *string* | **udi** }, enter **hardware-serial** to set the device serial number as the unique ID, enter **hostname** (the default) to select the device 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.
Step 16	**cns config initial** {*hostname* | *ip-address*} [*port-number*] [**event**] [**no-persist**] [**page** *page*] [**source** *ip-address*] [**syntax-check**] **Example:**	Enables the Cisco IOS agent, and initiates an initial configuration. • For {*hostname* | *ip-address*}, enter the hostname or the IP address of the configuration server.

	Command or Action	Purpose
	`RemoteDevice(config)# cns config initial 10.1.1.1 no-persist`	• (Optional) For *port-number*, 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 **page** *page*, 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. **Note** Though visible in the command-line help string, the **encrypt**, **status** *url*, and **inventory** keywords are not supported.
Step 17	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 18	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 19	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

What to do next

To verify information about the configuration agent, use the **show cns config connections** command in privileged EXEC mode.

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

Related Topics

Initial Configuration, on page 5
Monitoring CNS Configurations, on page 20

Refreshing DeviceIDs

Follow these steps to refresh a DeviceID when changing the hostname on the device.

SUMMARY STEPS

1. **enable**
2. **show cns config connections**
3. Make sure that the CNS event agent is properly connected to the event gateway.
4. **show cns event connections**
5. Record from the output of Step 4 the information for the currently connected connection listed below. You will be using the IP address and port number in subsequent steps of these instructions.
6. **configure terminal**
7. **no cns event** *ip-address port-number*
8. **cns event** *ip-address port-number*
9. **end**
10. Make sure that you have reestablished the connection between the device and the event connection by examining the output from **show cns event connections**.
11. **show running-config**
12. **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show cns config connections** **Example:** `Device# show cns config connections`	Displays whether the CNS event agent is connecting to the gateway, connected, or active, and the gateway used by the event agent, its IP address and port number.

	Command or Action	Purpose
Step 3	Make sure that the CNS event agent is properly connected to the event gateway.	Examine the output of **show cns config connections** for the following: • Connection is active. • Connection is using the currently configured device hostname. The DeviceID will be refreshed to correspond to the new hostname configuration using these instructions.
Step 4	**show cns event connections** **Example:** `Device# show cns event connections`	Displays the event connection information for your device.
Step 5	Record from the output of Step 4 the information for the currently connected connection listed below. You will be using the IP address and port number in subsequent steps of these instructions.	
Step 6	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 7	**no cns event** *ip-address port-number* **Example:** `Device(config)# no cns event 172.28.129.22 2012`	Specifies the IP address and port number that you recorded in Step 5 in this command. This command breaks the connection between the device and the event gateway. It is necessary to first break, then reestablish, this connection to refresh the DeviceID.
Step 8	**cns event** *ip-address port-number* **Example:** `Device(config)# cns event 172.28.129.22 2012`	Specifies the IP address and port number that you recorded in Step 5 in this command. This command reestablishes the connection between the device and the event gateway.
Step 9	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 10	Make sure that you have reestablished the connection between the device and the event connection by examining the output from **show cns event connections**.	
Step 11	**show running-config** **Example:**	Verifies your entries.

	Command or Action	Purpose
	`Device# show running-config`	
Step 12	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Related Topics

Hostname and DeviceID, on page 5

Enabling a Partial Configuration for Cisco IOS CNS Agent

Follow these steps to enable the Cisco IOS CNS agent and to initiate a partial configuration on the device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cns config partial** {*ip-address* | *hostname*} [*port-number*] [**source** *ip-address*]
4. **end**
5. **show running-config**
6. **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**cns config partial** {*ip-address* | *hostname*} [*port-number*] [**source** *ip-address*] **Example:**	Enables the configuration agent, and initiates a partial configuration.

	Command or Action	Purpose
	`Device(config)# cns config partial 172.28.129.22 2013`	• For {*ip-address* | *hostname*}, enter the IP address or the hostname of the configuration server. • (Optional) For *port-number*, 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. **Note** Though visible in the command-line help string, the **encrypt** keyword is not supported.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

What to do next

To verify information about the configuration agent, use either the **show cns config stats** or the **show cns config outstanding** command in privileged EXEC mode.

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** global configuration command.

Related Topics

Incremental (Partial) Configuration, on page 6
Monitoring CNS Configurations, on page 20

Monitoring CNS Configurations

Table 2: CNS show Commands

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

Related Topics

Enabling a Partial Configuration for Cisco IOS CNS Agent, on page 18
Incremental (Partial) Configuration, on page 6
Enabling an Initial Configuration for Cisco IOS CNS Agent, on page 11
Initial Configuration, on page 5