Configuring NTP

The Network Time Protocol (NTP) synchronizes the time of day among a set of distributed time servers and clients so that you can correlate events when you receive system logs and other time-specific events from multiple network devices. NTP uses the User Datagram Protocol (UDP) as its transport protocol. All NTP communications use Coordinated Universal Time (UTC).

Prerequisites for Configuring NTP

- NTP is disabled by default. The NTP IPv4 and IPv6 address must be configured.
- The controller must acquire the time from the NTP server.

Restrictions for Configuring NTP

- Maximum number of servers supported is 5.

Information About NTP

Network Time Protocol

Network Time Protocol (NTP) is a protocol designed to time-synchronize a network of machines. NTP runs on UDP, which in turn runs on IP. NTP Version 3 (NTPv3) is documented in RFC 1305.

An NTP network usually gets its time from an authoritative time source such as a radio clock or an atomic clock attached to a time server. NTP then distributes this time across the network. NTP is extremely efficient; no more than one packet per minute is necessary to synchronize two machines to the accuracy of within a millisecond of one another.

NTP uses the concept of a stratum to describe how many NTP hops away a machine is from an authoritative time source. A stratum 1 time server typically has an authoritative time source (such as a radio or atomic clock or a Global Positioning System [GPS] time source) directly attached, a stratum 2 time server receives its time via NTP from a stratum 1 time server, and so on.

NTP has two ways to avoid synchronizing to a machine whose time may not be accurate. NTP does not synchronize to a machine that is not in turn synchronized with the NTP. NTP compares the time reported by several machines and does not synchronize to a machine whose time is significantly different from others, even if its stratum is lower. This strategy effectively builds a self-organizing tree of NTP servers.

Our implementation of NTP does not support stratum 1 service; that is, you cannot connect to a radio or atomic clock (for some specific platforms, however, you can connect to a GPS time-source device). We recommend that the time service you derive for your network from the public NTP servers that are available in the IP Internet.

If the network is isolated from the Internet, our implementation of NTP allows a machine to be configured so that it acts as though it is synchronized via NTP, when in fact the network has determined the time by using other means. Other machines can then synchronize to that machine via NTP.

A number of manufacturers include NTP software for their host systems and a publicly available version for systems running UNIX. This software also allows UNIX-derivative servers to acquire the time directly from an atomic clock, which would subsequently propagate time information along to Cisco devices.

The communication between machines running NTP (known as associations) are usually statically configured; each machine is given the IP address of all machines with which it should form associations. Accurate timekeeping is made possible through exchange of NTP messages between each pair of machines with an association.

However, in a LAN environment, NTP can be configured to use IP broadcast messages instead. This alternative reduces configuration complexity because each machine can be configured to send or receive broadcast messages. However, the accuracy of timekeeping is marginally reduced because the information flow is only one way.

The time kept on a machine is a critical resource, so we strongly recommend that you use the security features of NTP to avoid the accidental or malicious setting of incorrect time. Two security mechanisms are available: an access-list-based restriction scheme and an encrypted authentication mechanism.

When multiple sources of time (VINES, hardware clock, manual configuration) are available, NTP is always considered to be more authoritative. NTP time overrides the time set by any other method.

NTP services are disabled on all interfaces by default.

For more information about NTP, see the following sections:

How to Configure NTP

Provisioning the Controller to Configure NTP

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 2	**controller nid** *1 NID_ID* **Example:** `Switch(config)# contoller nid 1/2`	Enters the controller configuration mode.
Step 3	**NtpPortType** **Example:** `Switch(config-controller)#` **`NtpPortType`**	Enters NTP provisioning mode.

Step 4	**NtpPortType** {**default** | **deleteNtpConfig** | **exit** | **getNtpConfig** | **no** | **setNtpConfig**}
	Example:
	`Switch(config-controller)#-NtpPortType)#` **`?`**
	`NtpPortType sub-mode commands:`
	`  default           Set a command to its defaults`
	`  deleteNtpConfig   delete NTP config request`
	`  exit              Exit from NtpPortType sub configuration`
	`mode`
	`  getNtpConfig      get ntp properties request`
	`  no                Negate a command or set its defaults`
	`  setNtpConfig      Set Ntp Server Details`
		Displays the supported configurations for NTP.

Step 5	**exit** **Example:** `Switch(config-controller-NtpPortType)#` **`exit`**	Exits the NTP provisioning mode.

Configuration Example

The following example shows the supported NTP configuration:

```
Switch(config-controller-NtpPortType)# ?
NtpPortType sub-mode commands:
  default          Set a command to its defaults
  deleteNtpConfig  delete NTP config request
  exit             Exit from NtpPortType sub configuration mode
  getNtpConfig     get ntp properties request
```

```
no                Negate a command or set its defaults
setNtpConfig      Set Ntp Server Details
```

Configuring NTP on the Controller

Before You Begin

- Ensure that the NID is reachable for the provided NTP server.
- Set the time zone for synchronization with the NTP server. See Configuring the System Clock.
- Perform the steps to provision NTP on the controller. See Provisioning the Controller to Configure NTP, on page 3

DETAILED STEPS

	Command or Action	Purpose

Step 1	**setNtpConfig** {**commit** | **flush** | **ntpConfig** | **review**}
	Example:
	`Switch(config-controller-NtpPortType)# `**`setNtpConfig ?`**
	`  commit            commit deleteNtpConfig`
	`  flush            flush all deleteNtpConfig commands from`
	` queue`
	`  ntpConfig         Set Ntp Server Details`
	`  review            review deleteNtpConfig commands`
		Sets NTP configuration
		• **commit**—Sends the NTP configuration to NID.
		• **flush**—Flushes all NTP configuration from the queue.
		• **ntpConfig**—Sets the NTP server configuration on the controller.
		• **review**—Displays the configuration on the controller.

Step 2	**setNtpConfig ntpConfig** {**hostinfo** {**hostname** *host_name*} | **ipv4address** *IPv4_address* | **ipv6address** *IPv6_address*} | **ntpmode** {**enable** | **number** *server_number*}}
	Example:
	`Switch(config-controller-NtpPortType)# `**`setNtpConfig`**
	`hostinfo hostname host1`
	`Switch(config-controller-NtpPortType)# `**`setNtpConfig`**
	`ipv4address 192.34.7.8`
	`Switch(config-controller-NtpPortType)# `**`setNtpConfig`**
	`ipv6address 2001:DB8:0:ABCD::1`
	`Switch(config-controller-NtpPortType)# `**`setNtpConfig ntpmode`**
	` enable`
	`Switch(config-controller-NtpPortType)# `**`setNtpConfig ntpmode`**
	` number 5`
		Configures NTP.
		• **hostinfo**—Sets the host information such as host name, IPv4 address and IPv6 address on the controller.
		• **ntpmode**—Enables or disables the NTP mode on the controller.
		• **number** *server_number*—Sets the NTP server details. The valid range is from 1 to 5.

Step 3	**setNtpconfig review**
	Example:
	`Switch(config-controller-NtpPortType)# `**`setNtpconfig review`**
	
	`Commands in queue:`
	`        setNtpConfig ntpConfig hostInfo hostName host1`
	`        setNtpConfig ntpConfig hostInfo ipv4Address`
		Displays the NTP configuration on the controller.

	Command or Action	Purpose

	`192.34.7.8`
	`        setNtpConfig ntpConfig ntpMode enable`
	`        setNtpConfig ntpConfig number 5`
	`        setNtpConfig ntpConfig ntpMode enable`	

Step 4	**setNtpconfigcommit** **Example:** `Switch(config-controller-NtpPortType)# ` **`setNtpconfig commit`**	Sends the NTP configuration to the NID.
Step 5	**exit** **Example:** `Switch(config-controller-NtpPortType)# ` **`exit`**	Exits the NTP provisioning mode.

Configuration Example

The example shows how to configure NTP on the controller:

```
Switch(config-controller-NtpPortType)# setNtpConfig hostinfo hostname host1
Switch(config-controller-NtpPortType)# setNtpConfig ipv4address 192.34.7.8
Switch(config-controller-NtpPortType)# setNtpConfig ipv6address 2001:DB8:0:ABCD::1
Switch(config-controller-NtpPortType)# setNtpConfig ntpmode enable
Switch(config-controller-NtpPortType)# setNtpConfig ntpmode number 5
Switch(config-controller-NtpPortType)# setNtpconfig review

Commands in queue:
        setNtpConfig ntpConfig hostInfo hostName host1
        setNtpConfig ntpConfig hostInfo ipv4Address 192.34.7.8
        setNtpConfig ntpConfig ntpMode enable
        setNtpConfig ntpConfig number 5
        setNtpConfig ntpConfig ntpMode enable

Switch(config-controller-NtpPortType)# setNtpconfig commit
Switch(config-controller-NtpPortType)# exit
```

Configuring NTP with Default Configuration

You can set the default NTP configuration on the controller.

Before You Begin

- Perform the steps to provision NTP on the controller. See Provisioning the Controller to Configure NTP, on page 3.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**default**{**getNtpConfig** | **setNtpConfig** | **deleteNtpConfig** | **exit** }	Sets the default NTP configuration.

	Command or Action	Purpose

	Example:
	`Switch(config-controller-NtpPortType)#` **`default ?`**
	
	`deleteNtpConfig   delete NTP config request`
	`exit              Exit from NtpPortType sub`
	`configuration mode`
	`getNtpConfig      get ntp properties request`
	`setNtpConfig      Set Ntp Server Details`
		- **getNtpConfig—**View the configuration on the controller.
		- **setNtpConfig—**Sets the configuration on the controller.
		- **deleteNtpConfig—**Deletes the configuration from the controller.
		- **exit—**Exits from NtpPortType configuration mode.

Step 2	**exit** **Example:** `Switch(config-controller-NtpPortType)#` **`exit`**	Exits the NTP provisioning mode.

Viewing the NTP Configuration

Before You Begin

- Perform the steps to provision NTP on the controller. See Provisioning the Controller to Configure NTP, on page 3.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**getNtpConfig** {**commit** | **flush** | **ntpStatusRequest** *ntp_status*| **review**}
	
	Example:
	
	`Switch(config-controller-NtpPortType)#`**`getNtpConfig ntpStatusRequest 1`**
	`Switch(config-controller-NtpPortType)#`**`getNtpConfig review`**
	`Switch(config-controller-NtpPortType)#`**`getNtpConfig commit`**
		- **ntpStatusRequest—**Request NTP configuration properties.
		- **commit—**Sends the NTP configuration to NID.
		- **flush—**Flushes all NTP configuration from the queue.
		- **review—**Displays the configuration.

Step 2	**exit** **Example:** `Switch(config-controller-NtpPortType)#` **`exit`**	Exits the NTP provisioning mode.

Configuration Example

The example shows how to view the configuration:

```
Switch(config-controller-NtpPortType)# getNtpConfig ntpStatusRequest 1
Switch(config-controller-NtpPortType)# getNtpConfig review

Commands in queue:
        getNtpConfig ntpStatusRequest 1
        getNtpConfig ntpStatusRequest 2
        getNtpConfig ntpStatusRequest 3

Switch(config-controller-NtpPortType)# getNtpConfig commmit
Switch(config-controller-NtpPortType)# end
```

Deleting the NTP Configuration

Before You Begin

- Perform the steps to provision NTP on the controller. See Provisioning the Controller to Configure NTP, on page 3.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**deleteNtpConfig** {**commit** | **flush** | **ntpDeleteConfig** | **review**}
	Example:
	`Switch(config-controller-NtpPortType)#` **`deleteNtpConfig ?`**
	`  commit            commit deleteNtpConfig`
	`  flush            flush all deleteNtpConfig commands`
	` from queue`
	`  ntpDeleteConfig   delete NTP config request`
	`  review            review deleteNtpConfig commands`
		Removes the NTP configuration.
		• **commit**—Sends the NTP configuration to NID.
		• **flush**—Flushes all NTP configuration from the queue.
		• **ntpDeleteConfig**—Deletes the NTP configuration request on the controller.
		• **review**—Displays the configuration on the controller.

Step 2	**ntpDeleteConfig** {**ntpEnable** | **ntpServerNo***server_num*}
	Example:
	`Switch(config-controller-NtpPortType)#`
	`deleteNtpConfig ntpDeleteConfig ntpEnable`
	`Switch(config-controller-NtpPortType)#`
	`deleteNtpConfig ntpDeleteConfig ntpServer 1`
		Removes NTP configuration.
		• **ntpEnable**—Disables the NTP configuration.
		• **ntpServerNo**—Disables the NTP server.
		• *server_num*—Specifies the NTP server. The valid range is from 1 to 5.

Step 3	**ntpDeleteConfig review**
	Example:
	`Switch(config-controller-NtpPortType)#`
	`deleteNtpConfig review`
		Displays the NTP configuration.

	Command or Action	Purpose

Step 4	**ntpDeleteConfig commit**
	Example:
	`Switch(config-controller-NtpPortType)#`
	`deleteNtpConfig commit`
		Sends the NTP configuration to the NID.

Step 5	**exit** **Example:** `Switch(config-controller-NtpPortType)#` **`exit`**	Exits the NTP provisioning mode.

Configuration Example

The following example shows how to delete the NTP configuration:

```
Switch(config-controller-NtpPortType)# deleteNtpConfig ntpDeleteConfig ntpEnable
Switch(config-controller-NtpPortType)# deleteNtpConfig ntpDeleteConfig ntpServer 1
Switch(config-controller-NtpPortType)# deleteNtpConfig review
Commands in queue:
        deleteNtpConfig ntpDeleteConfig ntpEnable
        deleteNtpConfig ntpDeleteConfig ntpServerNo 2
Switch(config-controller-NtpPortType)# deleteNtpConfig commit
DeleteNtpConfig Commit Success!!!
Switch(config-controller-NtpPortType)# deleteNtpConfig exit
```

Verifying NTP

Use these commands to verify the NTP status on the controller.

- **show ntp status**

  This command displays the NTP status on the NID. The following is a sample output from the command:

  ```
  Switch# show ntp status

  NTP Mode : disabled
  Idx   Server IP host address (a.b.c.d) or a host name string
  ---   ------------------------------------------------------
  1
  2
  3
  4
  5
  ```