



Configuring Network Time Protocol

- [Prerequisites for Implementing NTP on Cisco IOS XR Software, on page 1](#)
- [Information About Implementing NTP, on page 1](#)
- [How to Implement NTP, on page 3](#)
- [Configuration Examples for Implementing NTP, on page 19](#)

Prerequisites for Implementing NTP on Cisco IOS XR Software

You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Information About Implementing NTP

NTP synchronizes timekeeping among a set of distributed time servers and clients. This synchronization allows events to be correlated when system logs are created and other time-specific events occur.

NTP uses the User Datagram Protocol (UDP) as its transport protocol. All NTP communication uses Coordinated Universal Time (UTC). An NTP network usually receives its time from an authoritative time source, such as a radio clock or an atomic clock attached to a time server. NTP distributes this time across the network. NTP is extremely efficient; no more than one packet per minute is necessary to synchronize two machines to within a millisecond of each other.

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 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 avoids synchronizing to a machine whose time may not be accurate, in two ways. First, NTP never synchronizes to a machine that is not synchronized itself. Second, NTP compares the time reported by several machines and does not synchronize to a machine whose time is significantly different than the others, even if its stratum is lower. This strategy effectively builds a self-organizing tree of NTP servers.

The Cisco implementation of NTP does not support stratum 1 service; in other words, it is not possible to connect to a radio or atomic clock (for some specific platforms, however, you can connect a GPS time-source device). We recommend that time service for your network be derived from the public NTP servers available in the IP Internet.

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

Several manufacturers include NTP software for their host systems, and a publicly available version for systems running UNIX and its various derivatives is also available. 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 routers.

The communications 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 by exchanging NTP messages between each pair of machines with an association.

In a LAN environment, NTP can be configured to use IP broadcast messages. As compared to polling, IP broadcast messages reduce configuration complexity, because each machine can simply be configured to send or receive broadcast or multicast messages. However, the accuracy of timekeeping is marginally reduced because the information flow is one-way only.

An NTP broadcast client listens for broadcast messages sent by an NTP broadcast server at a designated IPv4 address. The client synchronizes the local clock using the first received broadcast message.

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 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.

Preventing Issues due to GPS Week Number Rollover (WNRO)

- If there are no GPS sources in the NTP source chain or server chain, there is no impact of GPS Week Number Rollover (WNRO).
- GPS WNRO affects only the system clock and not user traffic.
- Contact your GPS manufacturer to fix the GPS source for this condition.

To mitigate impact of GPS sources that are subject to GPS WNRO perform the following optional workarounds:

- If the GPS source has been identified to be a cause of potential disruption on April 6, 2019 (or after), configure `ntp master` in the Cisco that is device connected to this source, and its clock on the Stratum 1 device to preventively isolate it. This configuration enables the device to present its own clock for synchronization to downstream NTP clients.



Note The usage of `ntp master` command as mentioned above is only a workaround to this condition. Use this command until the GPS source-related conditions are resolved, and to prevent the distribution of incorrect clock values throughout the network.

- Configure multiple NTP servers (ideally 4, but more than 3) at Stratum 2 level of the network, to enable NTP clients at Stratum 2 level to get clock from more than one Stratum 1 server. This way, WNRO affected Stratum 1 servers are staged to be marked as 'false ticker' or 'outlier' clock sources as compared to other non-WNRO affected Stratum 1 servers.

How to Implement NTP

Configuring Poll-Based Associations



Note No specific command enables NTP; the first NTP configuration command that you issue enables NTP.

You can configure the following types of poll-based associations between the router and other devices (which may also be routers):

- Client mode
- Symmetric active mode

The client and the symmetric active modes should be used when NTP is required to provide a high level of time accuracy and reliability.

When a networking device is operating in the client mode, it polls its assigned time serving hosts for the current time. The networking device then picks a host from all the polled time servers to synchronize with. Because the relationship that is established in this case is a client-host relationship, the host does not capture or use any time information sent by the local client device. This mode is most suited for file-server and workstation clients that are not required to provide any form of time synchronization to other local clients. Use the **server** command to individually specify the time-serving hosts that you want your networking device to consider synchronizing with and to set your networking device to operate in the client mode.

When a networking device is operating in the symmetric active mode, it polls its assigned time-serving hosts for the current time and it responds to polls by its hosts. Because this is a peer-to-peer relationship, the host also retains time-related information about the local networking device that it is communicating with. This mode should be used when there are several mutually redundant servers that are interconnected via diverse network paths. Most stratum 1 and stratum 2 servers on the Internet today adopt this form of network setup. Use the **peer** command to individually specify the time-serving hosts that you want your networking device to consider synchronizing with and to set your networking device to operate in the symmetric active mode.

When the router polls several other devices for the time, the router selects one device with which to synchronize.



Note To configure a peer-to-peer association between the router and another device, you must also configure the router as a peer on the other device.

You can configure multiple peers and servers, but you cannot configure a single IP address as both a peer and a server at the same time.

To change the configuration of a specific IP address from peer to server or from server to peer, use the **no** form of the **peer** or **server** command to remove the current configuration before you perform the new configuration. If you do not remove the old configuration before performing the new configuration, the new configuration does not overwrite the old configuration.

Step 1 configure

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters mode.

Step 2 ntp**Example:**

```
RP/0/RP0/CPU0:router(config)# ntp
```

Enters NTP configuration mode.

Step 3 server ip-address [version number] [key key-id] [minpoll interval] [maxpoll interval] [source type interface-path-id] [prefer] [burst] [iburst]**Example:**

```
RP/0/RP0/CPU0:router(config-ntp)# server 172.16.22.44
minpoll 8 maxpoll 12
```

Forms a server association with another system. This step can be repeated as necessary to form associations with multiple devices.

Step 4 peer ip-address [version number] [key key-id] [minpoll interval] [maxpoll interval] [source type interface-path-id] [prefer]**Example:**

```
RP/0/RP0/CPU0:router(config-ntp)# peer 192.168.22.33
minpoll 8 maxpoll 12 source hundredGigE 0/0/0/1
```

Forms a peer association with another system. This step can be repeated as necessary to form associations with multiple systems.

Note To complete the configuration of a peer-to-peer association between the router and the remote device, the router must also be configured as a peer on the remote device.

Step 5 Use one of the following commands:

- **end**
- **commit**

Example:

```
RP/0/RP0/CPU0:router(config-ntp)# end
```

or

```
RP/0/RP0/CPU0:router(config-ntp)# commit
```

Saves configuration changes.

- When you issue the **end** command, the system prompts you to commit changes:

```
Uncommitted changes found, commit them before
  exiting(yes/no/cancel)?
[cancel]:
```

- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
 - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
 - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
- Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Broadcast-Based NTP Associates

In a broadcast-based NTP association, an NTP server propagates NTP broadcast packets throughout a network. Broadcast clients listen for the NTP broadcast packets propagated by the NTP server and do not engage in any polling.

Broadcast-based NTP associations should be used when time accuracy and reliability requirements are modest and if your network is localized and has a large number of clients (more than 20). Broadcast-based NTP associations also are recommended for use on networks that have limited bandwidth, system memory, or CPU resources. Time accuracy is marginally reduced in broadcast-based NTP associations because information flows only one way.

Use the **broadcast client** command to set your networking device to listen for NTP broadcast packets propagated through a network. For broadcast client mode to work, the broadcast server and its clients must be located on the same subnet. The time server that is transmitting NTP broadcast packets must be enabled on the interface of the given device using the **broadcast** command.

Use the **broadcast** command to set your networking device to send NTP broadcast packets.



Note No specific command enables NTP; the first NTP configuration command that you issue enables NTP.

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters mode.

Step 2 **ntp**

Example:

```
RP/0/RP0/CPU0:router(config)# ntp
```

Enters NTP configuration mode.

Step 3 (Optional) **broadcastdelay** *microseconds*

Example:

```
RP/0/RP0/CPU0:router(config-ntp)# broadcastdelay 5000
```

Adjusts the estimated round-trip delay for NTP broadcasts.

Step 4 **interface** *type interface-path-id***Example:**

```
RP/0/RP0/CPU0:router(config-ntp)# interface POS 0/1/0/0
```

Enters NTP interface configuration mode.

Step 5 **broadcast client****Example:**

```
RP/0/RP0/CPU0:router(config-ntp-int)# broadcast client
```

Configures the specified interface to receive NTP broadcast packets.

Note Go to next step to configure the interface to send NTP broadcast packets.

Step 6 **broadcast** [**destination** *ip-address*] [**key** *key-id*] [**version** *number*]**Example:**

```
RP/0/RP0/CPU0:router(config-ntp-int)# broadcast
destination 10.50.32.149
```

Configures the specified interface to send NTP broadcast packets.

Note Go to previous step to configure the interface to receive NTP broadcast packets.

Step 7 Use one of the following commands:

- **end**
- **commit**

Example:

```
RP/0/RP0/CPU0:router(config-ntp-int)# end
```

or

```
RP/0/RP0/CPU0:router(config-ntp-int)# commit
```

Saves configuration changes.

- When you issue the **end** command, the system prompts you to commit changes:

```
Uncommitted changes found, commit them before exiting(yes/no/cancel)?
[cancel]:
```

- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
- Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.

- Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
- Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring NTP Access Groups



Note No specific command enables NTP; the first NTP configuration command that you issue enables NTP.

The access list-based restriction scheme allows you to grant or deny certain access privileges to an entire network, a subnet within a network, or a host within a subnet.

The access group options are scanned in the following order, from least restrictive to most restrictive:

1. **peer**—Allows time requests and NTP control queries and allows the system to synchronize itself to a system whose address passes the access list criteria.
2. **serve**—Allows time requests and NTP control queries, but does not allow the system to synchronize itself to a system whose address passes the access list criteria.
3. **serve-only**—Allows only time requests from a system whose address passes the access list criteria.
4. **query-only**—Allows only NTP control queries from a system whose address passes the access list criteria.

If the source IP address matches the access lists for more than one access type, the first type is granted. If no access groups are specified, all access types are granted to all systems. If any access groups are specified, only the specified access types are granted.

For details on NTP control queries, see RFC 1305 (NTP version 3).

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters mode.

Step 2 **ntp**

Example:

```
RP/0/RP0/CPU0:router(config)# ntp
```

Enters NTP configuration mode.

Step 3 **access-group {peer | query-only | serve | serve-only} access-list-name**

Example:

```
RP/0/RP0/CPU0:router(config-ntp)# access-group peer access1
```

Creates an access group and applies a basic IPv4 or IPv6 access list to it.

Step 4 Use one of the following commands:

- **end**
- **commit**

Example:

```
RP/0/RP0/CPU0:router(config-ntp)# end
```

or

```
RP/0/RP0/CPU0:router(config-ntp)# commit
```

Saves configuration changes.

- When you issue the **end** command, the system prompts you to commit changes:

```
Uncommitted changes found, commit them before
  exiting(yes/no/cancel)?
[cancel]:
```

- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
 - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
 - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
- Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring NTP Authentication

This task explains how to configure NTP authentication.

The encrypted NTP authentication scheme should be used when a reliable form of access control is required. Unlike the access-list-based restriction scheme that is based on IP addresses, the encrypted authentication scheme uses authentication keys and an authentication process to determine if NTP synchronization packets sent by designated peers or servers on a local network are deemed as trusted, before the time information that it carries along is accepted.

The authentication process begins from the moment an NTP packet is created. A message authentication code (MAC) is computed using the MD5 Message Digest Algorithm and the MAC is embedded into an NTP synchronization packet. The NTP synchronization packet together with the embedded MAC and key number are transmitted to the receiving client. If authentication is enabled and the key is trusted, the receiving client computes the MAC in the same way. If the computed MAC matches the embedded MAC, the system is allowed to sync to the server that uses this key in its packets.

After NTP authentication is properly configured, your networking device only synchronizes with and provides synchronization to trusted time sources.

Step 1 **configure****Example:**

```
RP/0/RP0/CPU0:router# configure
Enters mode.
```

Step 2 **ntp****Example:**

```
RP/0/RP0/CPU0:router(config)# ntp
Enters NTP configuration mode.
```

Step 3 **authenticate****Example:**

```
RP/0/RP0/CPU0:router(config-ntp)# authenticate
Enables the NTP authentication feature.
```

Step 4 **authentication-key** *key-number* **md5** [**clear** | **encrypted**] *key-name***Example:**

```
RP/0/RP0/CPU0:router(config-ntp)# authentication-key 42
md5 clear key1
```

Defines the authentication keys.

- Each key has a key number, a type, a value, and, optionally, a name. Currently the only key type supported is **md5**.

Step 5 **trusted-key** *key-number***Example:**

```
RP/0/RP0/CPU0:router(config-ntp)# trusted-key 42
```

Defines trusted authentication keys.

- If a key is trusted, this router only synchronizes to a system that uses this key in its NTP packets.

Step 6 Use one of the following commands:

- **end**
- **commit**

Example:

```
RP/0/RP0/CPU0:router(config-ntp)# end
or
```

```
RP/0/RP0/CPU0:router(config-ntp)# commit
```

Saves configuration changes.

- When you issue the **end** command, the system prompts you to commit changes:

```
Uncommitted changes found, commit them before
  exiting(yes/no/cancel)?
[cancel]:
```

- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
 - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
 - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
- Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Disabling NTP Services on a Specific Interface

NTP services are disabled on all interfaces by default.

NTP is enabled globally when any NTP commands are entered. You can selectively prevent NTP packets from being received through a specific interface by turning off NTP on a given interface.

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
Enters mode.
```

Step 2 **ntp**

Example:

```
RP/0/RP0/CPU0:router(config)# ntp
Enters NTP configuration mode.
```

Step 3 Use one of the following commands:

- **no interface** *type interface-path-id*
- **interface** *type interface-path-id* **disable**

Example:

```
RP/0/RP0/CPU0:router(config-ntp)# no interface pos 0/0/0/1
```

or

```
RP/0/RP0/CPU0:router(config-ntp)# interface POS 0/0/0/1 disable
```

Disables NTP services on the specified interface.

Step 4 Use one of the following commands:

- **end**
- **commit**

Example:

```
RP/0/RP0/CPU0:router(config-ntp)# end
```

or

```
RP/0/RP0/CPU0:router(config-ntp)# commit
```

Saves configuration changes.

- When you issue the **end** command, the system prompts you to commit changes:

```
Uncommitted changes found, commit them before
  exiting(yes/no/cancel)?
[cancel]:
```

- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
 - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
 - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
- Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring the Source IP Address for NTP Packets

By default, the source IP address of an NTP packet sent by the router is the address of the interface through which the NTP packet is sent. Use this procedure to set a different source address.

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters mode.

Step 2 **ntp**

Example:

```
RP/0/RP0/CPU0:router(config)# ntp
```

Enters NTP configuration mode.

Step 3 *source type interface-path-id***Example:**

```
RP/0/RP0/CPU0:router(config-ntp)# source POS 0/0/0/1
```

Configures an interface from which the IP source address is taken.

Note This interface is used for the source address for all packets sent to all destinations. If a source address is to be used for a specific association, use the **source** keyword in the **peer** or **server** command shown in [Configuring Poll-Based Associations, on page 3](#).

Step 4 Use one of the following commands:

- **end**
- **commit**

Example:

```
RP/0/RP0/CPU0:router(config-ntp)# end
```

or

```
RP/0/RP0/CPU0:router(config-ntp)# commit
```

Saves configuration changes.

- When you issue the **end** command, the system prompts you to commit changes:

```
Uncommitted changes found, commit them before
  exiting(yes/no/cancel)?
[cancel]:
```

- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
 - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
 - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
- Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
-

Configuring the System as an Authoritative NTP Server

You can configure the router to act as an authoritative NTP server, even if the system is not synchronized to an outside time source.

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters mode.

Step 2 **ntp**

Example:

```
RP/0/RP0/CPU0:router(config)# ntp
```

Enters NTP configuration mode.

Step 3 **master *stratum***

Example:

```
RP/0/RP0/CPU0:router(config-ntp)# master 9
```

Makes the router an authoritative NTP server.

Note Use the **master** command with caution. It is very easy to override valid time sources using this command, especially if a low stratum number is configured. Configuring multiple machines in the same network with the **master** command can cause instability in time keeping if the machines do not agree on the time.

Step 4 Use one of the following commands:

- **end**
- **commit**

Example:

```
RP/0/RP0/CPU0:router(config-ntp)# end
```

or

```
RP/0/RP0/CPU0:router(config-ntp)# commit
```

Saves configuration changes.

- When you issue the **end** command, the system prompts you to commit changes:

```
Uncommitted changes found, commit them before
  exiting(yes/no/cancel)?
[cancel]:
```

- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.

- Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
- Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
- Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Updating the Hardware Clock

On devices that have hardware clocks (system calendars), you can configure the hardware clock to be periodically updated from the software clock. This is advisable for devices using NTP, because the time and date on the software clock (set using NTP) is more accurate than the hardware clock. The time setting on the hardware clock has the potential to drift slightly over time.

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters mode.

Step 2 **ntp**

Example:

```
RP/0/RP0/CPU0:router(config)# ntp
```

Enters NTP configuration mode.

Step 3 **update-calendar**

Example:

```
RP/0/RP0/CPU0:router(config-ntp)# update-calendar
```

Configures the router to update its system calendar from the software clock at periodic intervals.

Step 4 Use one of the following commands:

- **end**
- **commit**

Example:

```
RP/0/RP0/CPU0:router(config-ntp)# end
```

or

```
RP/0/RP0/CPU0:router(config-ntp)# commit
```

Saves configuration changes.

- When you issue the **end** command, the system prompts you to commit changes:

```
Uncommitted changes found, commit them before
  exiting(yes/no/cancel)?
[cancel]:
```

- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
 - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
 - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
- Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Verifying the Status of the External Reference Clock

This task explains how to verify the status of NTP components.



Note The commands can be entered in any order.

Step 1 **show ntp associations [detail] [location *node-id*]**

Example:

```
RP/0/RP0/CPU0:router# show ntp associations
```

Displays the status of NTP associations.

Step 2 **show ntp status [location *node-id*]**

Example:

```
RP/0/RP0/CPU0:router# show ntp status
```

Displays the status of NTP.

NTP-PTP Interworking

Table 1: Feature History Table

Feature Name	Release Information	Feature Description
NTP-PTP Interworking	Release 7.11.1	<p>We have improved NTP synchronization and reliability to achieve nanosecond-level accuracy for applications that require high-precision timing. This is achieved by enabling NTP-PTP interworking which allows the use of PTP as the reference clock.</p> <p>As in previous releases, the NTP client continues to support polling NTP protocol-based external time servers to synchronize the local system clock and achieve accuracy within the millisecond range.</p>

Before the support of NTP-PTP interworking, only backplane time was supported for the operating system time of the router.

Starting Cisco IOS XR Software Release 7.11.1, NTP-PTP interworking provides the ability to use PTP, and other valid time of day (TOD) sources such as Data over Cable Service Interface Specification (DOCSIS) Timing Interface (DTI) and global positioning system (GPS), as the time source for the operating system in the units of nanosec level accuracy. PTP is capable of achieving nanosecond-level accuracy, while NTP is typically only accurate to within milliseconds. By using PTP as a reference clock, NTP can improve its accuracy and meet the needs of applications that require high precision timing.

NTP-PTP interworking also provides the means to communicate status changes between PTP and NTP processes. It also supports the unambiguous control of the operating system time and backplane time in the event of bootup, switchovers, or card and process failures.

With NTP-PTP interworking, NTP is less likely to lose synchronization. As, PTP is more robust to network delays and disruptions than NTP. So, if there's a problem with the network, PTP can still maintain accurate synchronization.

Prerequisites for NTP-PTP Interworking

- Ensure that PTP is enabled, before configuring NTP-PTP Interworking.

For PTP, GM gets the clock from GPS/GNSS reference clock:

- If PTP-NTP feature is enabled on GM node, ensure GM gets clock reference from FPS/GNSS clock reference, config CLI on GM node.
- If PTP-NTP feature is enabled on BC node, ensure GM gets clock reference from FPS/GNSS clock reference, config CLI on BC Node.
- If PTP-NTP feature is enabled on TSC node, ensure GM gets clock reference from FPS/GNSS clock reference, and BC node gets the clock from that GM node, TSC node gets clock from BC node, and config CLI on TSC Node.

- If GM is not connected to any GPS/GNSS ref clock, default PTP is clock is set to Jan 1, 1970.

Enable NTP-PTP Interworking

You can configure NTP-PTP Interworking in any of the following ways:

- Setting NTP Primary Reference Clock as PTP

```
Router # Configure
Router(config) # ntp
Router(config-ntp) # master primary-reference-clock
Router(config-ntp) # commit
```

- Configuring NTP Server with IP address

The following example shows an NTP configuration to allow the system clock to be synchronized by time server hosts at IP address 198.51.100.1. You can take IP address of a neighbouring PTP interface.

```
Router # Configure terminal
Router(config) # ntp server 198.51.100.1
Router(config-ntp) # commit
```

Running Configuration

```
Router(config)#show running-config ntp
ntp
master primary-reference-clock
!
```

```
Router(config)#show running-config ntp
ntp
server 198.51.100.1
!
```

Verification

```
Router#show ntp status
```

```
Clock is synchronized, stratum 1, reference is 198.51.100.1
nominal freq is 1000000000.0000 Hz, actual freq is 101341889.2967 Hz, precision is 2**24
reference time is 8497CD13.A6AEB9DA (00:02:27.651 UTC Tue Jun 30 1970)
clock offset is -0.077 msec, root delay is 0.000 msec
root dispersion is 3937.89 msec, peer dispersion is 3937.74 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is 0.0000088676 s/s
system poll interval is 64, last update was 4 sec ago
authenticate is disabled, panic handling is disabled,
hostname resolution retry interval is 1440 minutes
```

```
Router#Show ntp associations
```

address	ref clock	st	when	poll	reach	delay	offset	disp
*~198.51.100.1	.PTP.	0	-	64	0	0.00	0.000	16000

FQDN for NTP Server

Table 2: Feature History Table

Feature Name	Release Information	Feature Description
FQDN for NTP Server on Non-default VRF	Release 7.9.1	<p>You can now specify a Fully Qualified Domain Name (FQDN) as the hostname for NTP server configuration over non-default VRFs.</p> <p>FQDNs are easy to remember compared to numeric IP addresses. Service migration from one host to another can cause a change in IP address leading to outages.</p> <p>Prior releases allowed FQDN handling in only default VRFs.</p>

NTP on Cisco IOS XR Software supports configuration of servers and peers using their Fully Qualified Domain Names (FQDN). While configuring, the FQDN is resolved via DNS into its corresponding IPv4 or IPv6 address and is stored in the running-configuration of the system. NTP supports FQDN for both IPv4 and IPv6 protocols. You can configure FQDN on default vrf.

Starting Cisco IOS XR Software Release 7.9.1 you can configure FQDN in nondefault vrf also.

Configure FQDN for NTP server

Configuration Example for FQDN on NTP Server on Default VRF

Use the **ntp server** command with the FQDN name to configure FQDN on default VRF. You dont need to specify VRF name. In the following example, *time.cisco.com* is the FQDN.

```
Router#configure
Router(config)#ntp server time.cisco.com
Router(config)#commit
```

Running Configuration

Use the **show running-config ntp** command to see the ntp running configuration.

```
Router#show running-config ntp
ntp
 server 192.0.2.1
!
```

Verification

Use the **show ntp associations** command to verify that an NTP association has come up.

```
Router#show ntp associations

      address      ref clock      st  when  poll reach  delay  offset  disp
~192.0.2.1      173.38.201.67   2   42   128   3  196.06  -14.25  3949.4
* sys_peer, # selected, + candidate, - outlayer, x falseticker, ~ configured
```

Prerequisites for configuring FQDN in a nondefault VRF

- Configuration must exist for DNS resolution over that specific VRF.
- The server must be reachable.

Configuration Example for FQDN on NTP Server on Nondefault VRF

FQDN must be reachable from the router to configure it as an NTP server or peer. You can use the **ping** command and verify that FQDN is reachable. In the following example, *time.cisco.com* is the FQDN and *vrf_1* is the VRF over which it is reachable.

```
Router#ping time.cisco.com vrf vrf_1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.0.2.1 timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 171/171/172 ms
```

When you have confirmed that FQDN is reachable, you can configure FQDN to be used as an NTP server/peer.

```
Router#configure
Router(config)#ntp server vrf vrf_1 time.cisco.com minpoll 4 maxpoll 4 iburst
Router(config)#commit
```



Note If the FQDN you're trying to configure isn't reachable, the CLI treats it as invalid input.

Running Configuration

Use the **show running-config ntp** command to see the ntp running configuration.

```
Router#show running-config ntp
ntp
server vrf vrf_1 192.0.2.1 minpoll 4 maxpoll 4 iburst
!
```

Verification

Use the **show ntp associations** command to verify that an NTP association has come up.

```
Router#show ntp associations
address          ref clock      st  when  poll reach  delay  offset  disp
~192.0.2.1 vrf vrf_1
                  173.38.201.115 2    14   16   37  179.10 13.492 16.680
* sys_peer, # selected, + candidate, - outlayer, x falseticker, ~ configured
```

Configuration Examples for Implementing NTP

Configuring Poll-Based Associations: Example

The following example shows an NTP configuration in which the router's system clock is configured to form a peer association with the time server host at IP address 192.168.22.33, and to allow the system clock to be synchronized by time server hosts at IP address 10.0.2.1 and 172.19.69.1:

```
ntp
 server 10.0.2.1 minpoll 5 maxpoll 7
 peer 192.168.22.33

 server 172.19.69.1
```

Configuring Broadcast-Based Associations: Example

The following example shows an NTP client configuration in which interface 0/2/0/0 is configured to receive NTP broadcast packets, and the estimated round-trip delay between an NTP client and an NTP broadcast server is set to 2 microseconds:

```
ntp
 interface hundredGigE 0/2/0/0
   broadcast client
 exit
 broadcastdelay 2
```

The following example shows an NTP server configuration where interface 0/2/0/2 is configured to be a broadcast server:

```
ntp
 interface hundredGigE 0/2/0/2
   broadcast
```

Configuring NTP Access Groups: Example

The following example shows a NTP access group configuration where the following access group restrictions are applied:

- Peer restrictions are applied to IP addresses that pass the criteria of the access list named peer-acl.
- Serve restrictions are applied to IP addresses that pass the criteria of access list named serve-acl.
- Serve-only restrictions are applied to IP addresses that pass the criteria of the access list named serve-only-acl.
- Query-only restrictions are applied to IP addresses that pass the criteria of the access list named query-only-acl.

```
ntp
 peer 10.1.1.1
 peer 10.1.1.1
 peer 10.2.2.2
 peer 10.3.3.3
 peer 10.4.4.4
 peer 10.5.5.5
 peer 10.6.6.6
 peer 10.7.7.7
 peer 10.8.8.8
 access-group peer peer-acl
 access-group serve serve-acl
```

```

access-group serve-only serve-only-acl
access-group query-only query-only-acl
exit
ipv4 access-list peer-acl
 10 permit ip host 10.1.1.1 any
 20 permit ip host 10.8.8.8 any
exit
ipv4 access-list serve-acl
 10 permit ip host 10.4.4.4 any
 20 permit ip host 10.5.5.5 any
exit
ipv4 access-list query-only-acl
 10 permit ip host 10.2.2.2 any
 20 permit ip host 10.3.3.3 any
exit
ipv4 access-list serve-only-acl
 10 permit ip host 10.6.6.6 any
 20 permit ip host 10.7.7.7 any
exit

```

Configuring NTP Authentication: Example

The following example shows an NTP authentication configuration. In this example, the following is configured:

- NTP authentication is enabled.
- Two authentication keys are configured (key 2 and key 3).
- The router is configured to allow its software clock to be synchronized with the clock of the peer (or vice versa) at IP address 10.3.32.154 using authentication key 2.
- The router is configured to allow its software clock to be synchronized with the clock by the device at IP address 10.32.154.145 using authentication key 3.
- The router is configured to synchronize only to systems providing authentication key 3 in their NTP packets.

```

ntp
authenticate
authentication-key 2 md5 encrypted 06120A2D40031D1008124
authentication-key 3 md5 encrypted 1311121E074110232621
trusted-key 3
server 10.3.32.154 key 3
peer 10.32.154.145 key 2

```

Disabling NTP on an Interface: Example

The following example shows an NTP configuration in which 0/2/0/0 interface is disabled:

```

ntp
interface hundredGigE 0/2/0/0
disable
exit
authentication-key 2 md5 encrypted 06120A2D40031D1008124

```

```
authentication-key 3 md5 encrypted 1311121E074110232621
authenticate
trusted-key 3
server 10.3.32.154 key 3
peer 10.32.154.145 key 2
```

Configuring the Source IP Address for NTP Packets: Example

The following example shows an NTP configuration in which Ethernet management interface 0/0/CPU0/0 is configured as the source address for NTP packets:

```
ntp
authentication-key 2 md5 encrypted 06120A2D40031D1008124
authentication-key 3 md5 encrypted 1311121E074110232621
authenticate
trusted-key 3
server 10.3.32.154 key 3
peer 10.32.154.145 key 2
source MgmtEth0/0/CPU0/0
```

Configuring the System as an Authoritative NTP Server: Example

The following example shows a NTP configuration in which the router is configured to use its own NTP master clock to synchronize with peers when an external NTP source becomes unavailable:

```
ntp
master 6
```

Updating the Hardware Clock: Example

The following example shows an NTP configuration in which the router is configured to update its hardware clock from the software clock at periodic intervals:

```
ntp
server 10.3.32.154
update-calendar
```