

# Novell Open Enterprise Server

#.#

February 14, 2005

NETWORK TIME PROTOCOL  
ADMINISTRATION GUIDE

[www.novell.com](http://www.novell.com)



Novell®

## Legal Notices

Novell, Inc. makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc. makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

You may not use, export, or re-export this product in violation of any applicable laws or regulations including, without limitation, U.S. export regulations or the laws of the country in which you reside.

Copyright © 2004 Novell, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the publisher.

U.S. Patent No. 5,157,663; 5,349,642; 5,455,932; 5,553,139; 5,553,143; 5,572,528; 5,594,863; 5,608,903; 5,633,931; 5,652,854; 5,671,414; 5,677,851; 5,692,129; 5,701,459; 5,717,912; 5,758,069; 5,758,344; 5,781,724; 5,781,733; 5,784,560; 5,787,439; 5,818,936; 5,828,882; 5,832,274; 5,832,275; 5,832,483; 5,832,487; 5,859,978; 5,864,865; 5,870,561; 5,870,739; 5,873,079; 5,878,415; 5,884,304; 5,893,118; 5,903,650; 5,903,720; 5,905,860; 5,910,803; 5,913,025; 5,913,209; 5,915,253; 5,919,257; 5,925,108; 5,933,503; 5,933,602; 5,933,826; 5,946,002; 5,946,467; 5,956,718; 5,956,745; 5,964,872; 5,974,474; 5,983,223; 5,983,234; 5,987,471; 5,991,810; 6,002,398; 6,014,667; 6,015,132; 6,016,499; 6,023,586; 6,029,247; 6,047,289; 6,047,312; 6,052,724; 6,061,726; 6,061,740; 6,061,743; 6,065,017; 6,067,093; 6,081,774; 6,081,814; 6,081,900; 6,094,672; 6,098,090; 6,105,062; 6,105,069; 6,105,132; 6,108,649; 6,115,039; 6,119,122; 6,144,959; 6,151,688; 6,157,925; 6,167,393; 6,173,289; 6,216,123; 6,219,652; 6,233,859; 6,247,149; 6,269,391; 6,286,010; 6,308,181; 6,314,520; 6,324,670; 6,338,112; 6,345,266; 6,353,898; 6,424,976; 6,466,944; 6,477,583; 6,477,648; 6,484,186; 6,496,865; 6,510,450; 6,516,325; 6,519,610; 6,532,451; 6,532,491; 6,539,381; 6,560,615; 6,567,873; 6,578,035; 6,591,397; 6,609,158; 6,615,350; 6,629,105; 6,629,132; 6,647,408; 6,651,242; 6,671,688; 6,701,433; 6,728,959; 6,742,035; 6,751,735; 6,775,668; RE37,178. Patents Pending.

Novell, Inc.  
404 Wyman Street, Suite 500  
Waltham, MA 02451  
U.S.A.

[www.novell.com](http://www.novell.com)

Novell Network Time Protocol Administration Guide for OES  
[December 15, 2004](#)

**Online Documentation:** To access the online documentation for this and other Novell products, and to get updates, see [www.novell.com/documentation](http://www.novell.com/documentation).

**Novell Trademarks**

Novell is a registered trademark of Novell, Inc. in the United States and other countries.  
SUSE is a registered trademark of SUSE AG, a Novell business.

**Third-Party Materials**

All third-party trademarks are the property of their respective owners.



# Contents

<b>About this Guide</b>	<b>7</b>
<b>1 Network Time Protocol</b>	<b>9</b>
NTP Terminology . . . . .	9
NTP Implementation on NetWare . . . . .	11
NTP Architecture . . . . .	11
Compatibility with Other Versions of NetWare . . . . .	12
<b>2 Modes of Time Synchronization</b>	<b>13</b>
Client-Server Mode . . . . .	13
Peer-to-Peer Mode . . . . .	14
Broadcast and Multicast Mode . . . . .	14
<b>3 NTP Configuration</b>	<b>17</b>
Manual Configuration . . . . .	17
Planning the Setup . . . . .	17
Configuring the Time Providers . . . . .	18
Configuring the Time Consumers . . . . .	19
Sample Scenario . . . . .	19
Auto Configuration . . . . .	20
Planning the Setup . . . . .	21
Configuring the Time Provider . . . . .	21
Configuring the Time Consumers . . . . .	21
Sample Scenario . . . . .	21
Wide Area Configuration . . . . .	22
Planning the Setup . . . . .	22
Configuring the Time Provider Group . . . . .	22
Configuring the Time Consumers . . . . .	23
Sample Scenario . . . . .	24
Synopsis . . . . .	25
Remote Configuration . . . . .	26
<b>4 Migrating TimeSync Servers to NTP</b>	<b>27</b>
Reviewing Pre-requisites . . . . .	27
Selecting the Mode of Migration . . . . .	27
Automatically Migrating Servers . . . . .	28
Manually Migrating Servers . . . . .	29
Configuring Servers Running the NTP Services . . . . .	30
Monitoring the Health Status of Migrated Servers . . . . .	32
<b>5 NTP Utilities</b>	<b>33</b>
NTPDate . . . . .	33
NTPQ . . . . .	34
Internal Commands . . . . .	35
Control Message Commands . . . . .	36
NTPTrace . . . . .	38

XNTPDC . . . . .	39
XNTPD . . . . .	40
The Configuration File . . . . .	41
Files . . . . .	44
Configuration Options . . . . .	44
Authentication Options . . . . .	47
Monitoring Options . . . . .	49
Access Control Options . . . . .	51
Miscellaneous Options . . . . .	52
<b>6 Monitoring and Security</b>	<b>53</b>
Monitoring Time Synchronization . . . . .	53
Health Monitor . . . . .	55
Security . . . . .	55
Sample Scenario . . . . .	57
<b>7 Troubleshooting NTP</b>	<b>61</b>
XNTPD . . . . .	61
NTPDate . . . . .	63
General . . . . .	63
<b>8 Frequently Asked Questions</b>	<b>65</b>
How Do I Configure NTP for Fault Tolerance? . . . . .	65
How Do I Find the Subnet Broadcast Address? . . . . .	66
How Do I Find which Server Is Selected for Synchronization? . . . . .	66
How Do I Find the Offset Value between the Client and the Server? . . . . .	66
Can a Single Timesync Server Take Time from a NetWare 6.5 Server Running NTPv3? . . . . .	66
<b>Revision History</b>	<b>67</b>
May 9, 2005 . . . . .	67

## About this Guide

This guide describes the time synchronization service based on RFC 1305. Time synchronization uses Network Time Protocol version 3 (NTPv3). This guide is intended to help network administrators configure and use NTPv3, and it contains the following sections:

- ♦ Chapter 1, “Network Time Protocol,” on page 9
- ♦ Chapter 2, “Modes of Time Synchronization,” on page 13
- ♦ Chapter 3, “NTP Configuration,” on page 17
- ♦ Chapter 5, “NTP Utilities,” on page 33
- ♦ Chapter 6, “Monitoring and Security,” on page 53
- ♦ Chapter 7, “Troubleshooting NTP,” on page 61
- ♦ Chapter 8, “Frequently Asked Questions,” on page 65

### Documentation Updates

For the most recent version of this *Network Time Protocol Administration Guide*, see the [Novell documentation Web site \(http://www.novell.com/documentation/lg/nw65\)](http://www.novell.com/documentation/lg/nw65).

### Documentation Conventions

All occurrences of NTP in this documentation refer to NTP version 3 (NTPv3).

In this documentation, a greater-than symbol (>) is used to separate actions within a step and items in a cross-reference path.

Also, a trademark symbol (®, ™, etc.) denotes a Novell trademark. An asterisk (\*) denotes a third-party trademark.

When a single pathname can be written with a backslash for some platforms or a forward slash for other platforms, the pathname is presented with a backslash. Users of platforms that require a forward slash, such as UNIX\*, should use forward slashes as required by your software.





# 1

## Network Time Protocol

The Network Time Protocol (NTP) is used to synchronize servers, which are NTPv3 compliant. Operating systems like NetWare®, Linux\*, and Solaris\* are NTPv3 compliant.

For more information on NTP, refer to the following Web sites:

- ♦ [www.ntp.org](http://www.ntp.org) (<http://www.ntp.org>)
- ♦ [www.eecis.udel.edu/~mills](http://www.eecis.udel.edu/~mills) (<http://www.eecis.udel.edu/~mills>)

This chapter explains the following:

- ♦ [NTP Terminology \(page 9\)](#)
- ♦ [NTP Implementation on NetWare \(page 11\)](#)
- ♦ [NTP Architecture \(page 11\)](#)
- ♦ [Compatibility with Other Versions of NetWare \(page 12\)](#)

## NTP Terminology

This documentation uses the following terms:

- ♦ **NTP time provider**

A server that understands the Network Time Protocol (NTP) protocol and provides NTP time to other servers or to workstations on the network.

The time provider gives time to operating systems that are NTP and NCP compliant, such as,

- ♦ NetWare 4.2, 5.0, 5.1, 6.0, and 6.5
- ♦ All flavors of UNIX
- ♦ All versions of Windows that have NTP compliance

The time provider can broadcast or multicast its services on the network. For more information, see [“Broadcast and Multicast Mode” on page 14](#).

- ♦ **NTP time consumer**

A server that understands the NTP protocol and seeks NTP time from an NTP time provider to synchronize its time.

The time consumer can accept a time provider in the client-server, peer-peer, and multicast/broadcast modes. For more information, see [“Client-Server Mode” on page 13](#), [“Peer-to-Peer Mode” on page 14](#), and [“Broadcast and Multicast Mode” on page 14](#).

The time consumer can work with operating systems that are NTP and NCP compliant, like:

- ♦ NetWare 5.1, 6.0, and 6.5
- ♦ All flavors of UNIX

- ♦ All versions of Windows that have NTP compliance
- ♦ **Time provider group**

A set of servers that are configured to ensure fault tolerance and optimal network usage.

The time provider group can be configured to keep the network traffic at a minimum.
- ♦ **Dispersion**

A measure (in seconds) of how scattered the time offsets are from a given time server.
- ♦ **Drift**

A measure (in Hertz per second) of how quickly the skew of a clock changes. See also “**Slew**” on page 10.
- ♦ **Jitter**

Small rapid variations in a waveform due to fluctuations in the voltage supply, mechanical vibrations, or other sources.
- ♦ **Minpoll**

Specifies the minimum polling interval (in seconds to the power of 2) for NTP messages. If you set minpoll to 4, the minimum polling interval reduces. A value of 4 with minpoll helps the server to synchronize within a minute.

The default minpoll value is 4 on NetWare and 6 on UNIX.
- ♦ **Root delay**

The total roundtrip delay (in seconds) to the primary reference source at the root of the synchronization subnet. This variable can take on both positive and negative values, depending on clock precision and skew.
- ♦ **Root dispersion**

The maximum error (in seconds) relative to the primary reference source at the root of the synchronization subnet. This variable can take on only positive values greater than zero.
- ♦ **Skew**

A measure (in hertz) of the difference between the actual frequency of a clock and its frequency to keep perfect time. See also “**Drift**” on page 10.
- ♦ **Slam**

To immediately correct or adjust the time of a clock. This might lead to sudden bursts in time.
- ♦ **Step**

To change the time of a clock to the correct time with no intermediate adjustments. See “**Skew**” on page 10.
- ♦ **Slew**

To gradually adjust the time of a clock until it displays the correct time. See “**Step**” on page 10.
- ♦ **Stratum**

Conventions established to indicate the accuracy of each time server and is defined by a number called the stratum, with the topmost level (primary servers) assigned as one and each level downwards (secondary servers) in the hierarchy assigned as one greater than the preceding level.

## NTP Implementation on NetWare

Novell® eDirectory™ 8.7.3 relies heavily on consistent and reliable time stamps for its objects. Without such time stamps, synchronization of directories is not possible.

By default, Timesync is loaded with NetWare. To make XNTPD load by default, edit the `sys:\system\timeserv.ncf` file.

NTP addresses fault tolerance in the form of a time provider group. In a time provider group, all the servers in one geographical location network obtain time from other servers in the same network. Only one server communicates with a server outside the network and obtains time from it. Therefore, the network traffic across the geographical locations reduces, minimizing traffic across routers and WANs.

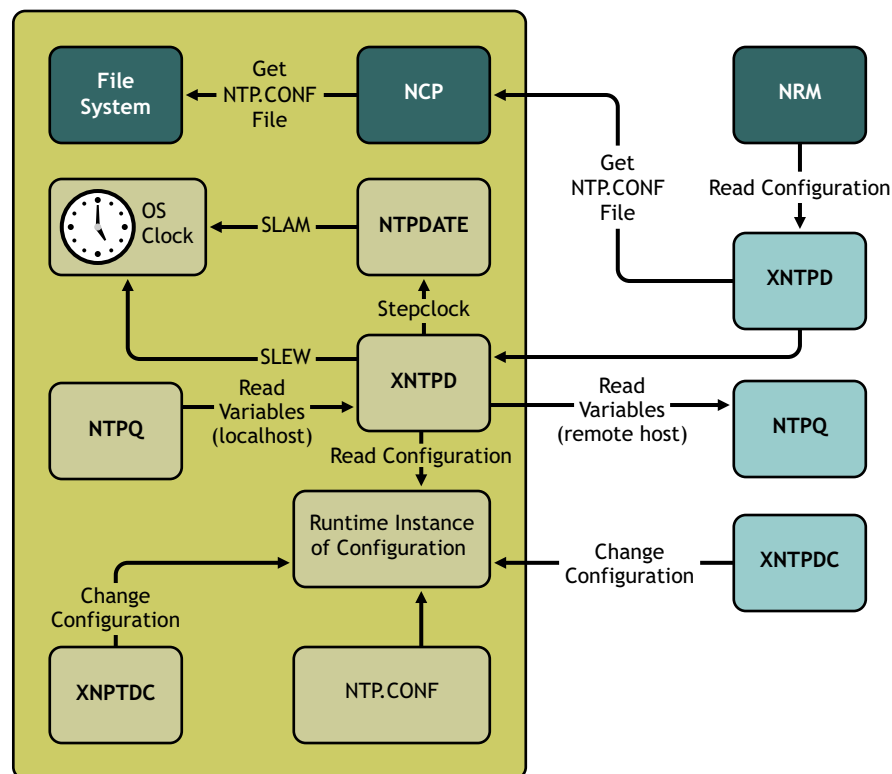
### Features of NTP on NetWare

- ♦ All features as per RFC 1305 on NTPv3.
- ♦ Support for browser-based configuration (NORM). This helps in centralizing the support and configuration of NTP time synchronization service on the network for the complete eDirectory tree.
- ♦ Support of backward compatibility for Timesync.

## NTP Architecture

This section briefly outlines the NTP architecture.

Figure 1 NTP Architecture



The above figure illustrates the following:

- ♦ **NTPDate** sets the local time and date.
- ♦ **NTPQ** queries the status or quality of time parameters.
- ♦ **NTPTrace** queries the time server and its servers until the master server is queried. NTPTrace determines where a given NTP server gets its time from, then follows the chain of NTP servers back to their master time source.
- ♦ **XNTPDC** is the remote configuration utility. It is used to query the XNTPD daemon about its current state and to request changes in that state.
- ♦ **XNTPD** is an operating system daemon which sets and maintains the system time-of-day in synchronism with Internet standard time servers.

The components within the box relate to a particular server.

NTPDate and XNTPD communicate with the server's XNTPD, which is trying to synchronize the time.

NTPDate gets the time from another server and slams the time on the local clock. Slamming the time immediately overwrites the time on the local clock.

XNTPD gets the time from another server and slews the time on the local clock. Slewing the time adjusts the local clock to the time of the other server gradually. XNTPD sets and maintains the time on the local clock.

Ntp.conf is the configuration file. XNTPD reads this file at startup in order to determine the synchronization sources and operating modes. The time configuration values are entered in the ntp.conf file.

## Compatibility with Other Versions of NetWare

NTPv3 components cannot be loaded on a server where timesync.nlm is running.

All versions of NetWare 4 can talk only NCP™. NetWare 5.x and 6.x can talk NCP and NTP.

The NTPv3 component, XNTPD, replies to NCP time requests to support backward compatibility for Timesync.

Time Provider	Time Consumer	Is Configuration Allowed?
Timesync SINGLE time server	NTP client time consumer	Allowed - NTP
Timesync PRIMARY time server	NTP client time consumer	Allowed - NTP
Timesync SECONDARY time server	NTP client time consumer	Allowed - NTP
NTP server time provider	SINGLE time server	Allowed - NCP/NTP
	REFERENCE time server	Allowed - NCP/NTP
	PRIMARY time server	Allowed - NCP/NTP
	SECONDARY time server	Allowed - NCP/NTP
	NTP client Time Consumer	NTP

# 2

## Modes of Time Synchronization

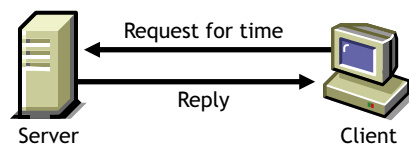
You can synchronize time using the following three methods:

- ♦ [Client-Server Mode \(page 13\)](#)
- ♦ [Peer-to-Peer Mode \(page 14\)](#)
- ♦ [Broadcast and Multicast Mode \(page 14\)](#)

### Client-Server Mode

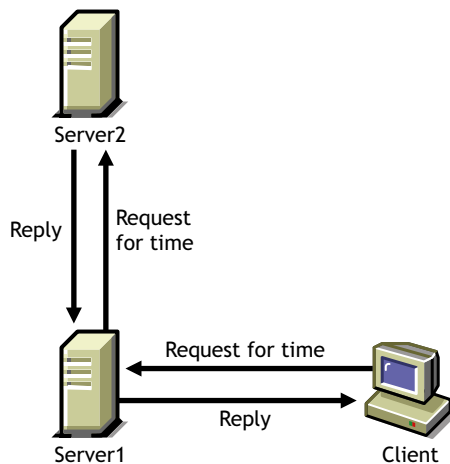
With this mode, the time consumer requests the time provider for the time and the time provider sends back the time taking into account the time delays and other contingencies.

**Figure 2 Client-Server Mode**



The time provider might also be a time consumer to another time provider. This scenario is displayed in the next figure. In this case, the time provider (Server1) requests the time from its time provider (Server2) and, upon getting a reply, responds to the time consumer (Client).

**Figure 3 Time Provider As a Time Consumer**



You must synchronize the server prior to synchronizing the client. Therefore, it is important to configure the server in either of the following ways:

- ♦ Self-synchronized

Here, the local clock is used as the time source and the server synchronizes with it. Add the following lines to the `ntp.conf` file (located in `sys:\etc`):

```
server 127.127.1.0

fudge server 127.127.1.0 stratum 3
```

- ♦ As a client to another server

Here, the server is configured as a time consumer to another time provider. Add the following line in the `ntp.conf` file:

```
server IP_address_of_time_provider
```

After you configure the server, you can configure the clients to use this server as a time provider. To do this, add the following lines to the `ntp.conf` file:

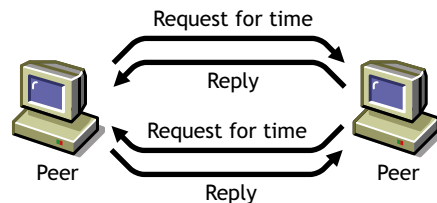
```
server IP_address_of_server
```

See [“Manual Configuration” on page 17](#) for more information.

## Peer-to-Peer Mode

With this mode, there are two time consumers or time providers. Both of them can request the time from each other and respond to each other. They are at the same level and, therefore, known as peers.

**Figure 4 Peer-to-Peer Mode**



If you want to use the servers in this mode, add the following lines to the `ntp.conf` file of each server:

```
peer IP_address_of_the_other_server
```

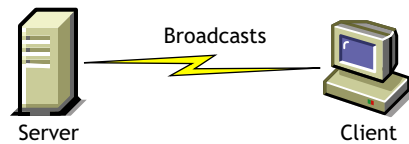
See [“Wide Area Configuration” on page 22](#) for more information.

## Broadcast and Multicast Mode

With the broadcast or multicast mode, the time provider broadcasts (or multicasts) its service within the subnet. The time consumer listens to the broadcast (or multicast) and registers it as its time provider. The time consumer then requests the time from the time provider.

This mode helps to avoid reconfiguring the entire network if the time provider of the network changes.

To change the time provider, you must remove the time provider from the network and replace it with another to broadcast (or multicast) its service.

**Figure 5 Broadcast/Multicast Mode**

To configure servers in the broadcast or multicast mode, you must have a time provider advertising its service. The time consumers that need to obtain time from this time provider would listen to the advertisement, register for the service, and use it.

### Broadcast

To make the time provider broadcast its service, add the following line to its `ntp.conf` file:

```
broadcast subnet_broadcast_address key key_ID
```

To make the time consumer listen to the services that are broadcast, add the following line to its `ntp.conf` file:

```
broadcastclient subnet_broadcast_address
```

**NOTE:** The *subnet broadcast address* variable is optional.

### Multicast

To make the time provider multicast its service, add the following line to its `ntp.conf` file:

```
broadcast 224.0.1.1 key key_ID
```

To make the time consumer listen to the services that are multicast, add the following line to its `ntp.conf` file:

```
multicastclient
```

See [“Auto Configuration” on page 20](#) for more details.





# 3

## NTP Configuration

You can configure NTPv3 as a replacement for timesync.nlm.

Prior to configuring NTP, it is important to understand the different types of time synchronization methods. The NTP configuration is based on the synchronization type you choose to use.

There are three types of configuration methods:

- ♦ [Manual Configuration \(page 17\)](#)
- ♦ [Auto Configuration \(page 20\)](#)
- ♦ [Wide Area Configuration \(page 22\)](#)

It is important to understand when to use a particular type of configuration. The following table explains the type of configuration you can select for each mode of synchronization.

Configuration Type	Synchronization Mode
Manual	<ul style="list-style-type: none"><li>♦ Client-server</li><li>♦ Self-synchronized</li></ul>
Auto	Broadcast/Multicast
Wide Area	<ul style="list-style-type: none"><li>♦ Peer-to-Peer</li><li>♦ A combination of modes</li></ul>

### Manual Configuration

Manual configuration is easy to plan, configure, and debug. This type of configuration is best suited in a setup where there are fewer than 15 servers and the servers are in the same geographical location and do not span across a large WAN.

You can manually configure a time synchronized setup by completing the following tasks:

1. [Planning the Setup.](#)
2. [Configuring the Time Providers.](#)
3. [Configuring the Time Consumers.](#)

### Planning the Setup

You should plan the setup before configuring the time provider and time consumers. The setup consists of a time group, which is a set of servers synchronized for time.

The plan should include the following:

Identify the most reliable server in the subnet and make it the time provider.  
Identify the other servers in the subnet be the time consumers.

## Configuring the Time Providers

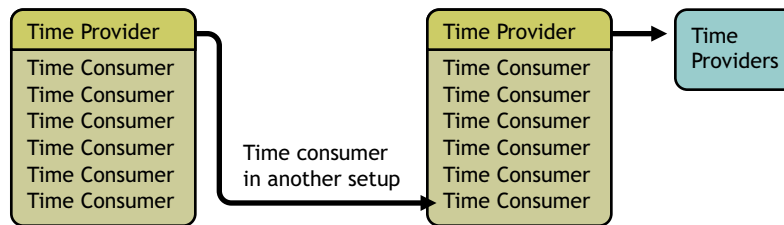
In manual configuration, a time provider can get time

- ♦ From another time provider (client-server mode)
- ♦ From its local clock (self-synchronized mode)

### Client-Server Mode

In this mode, a time consumer takes time from a time provider. The time provider may be a time consumer in another setup or may take time from an external time provider as shown in the figure below.

**Figure 6 Time Group Taking Time from a Time Provider**



To configure the time provider:

- 1** Add a line similar to the following to the time provider's `ntp.conf` file:

```
server IP_address_of_time_provider prefer
```

The *prefer* parameter marks the server as preferred. All other things being equal, this time provider is chosen for synchronization among a set of correctly operating providers.

`ntp.conf` file is the configuration file for NTP. This file is located in `sys:\etc`. For more information, see [“The Configuration File” on page 41](#).

- 2** Load XNTPD for the changes to take effect.

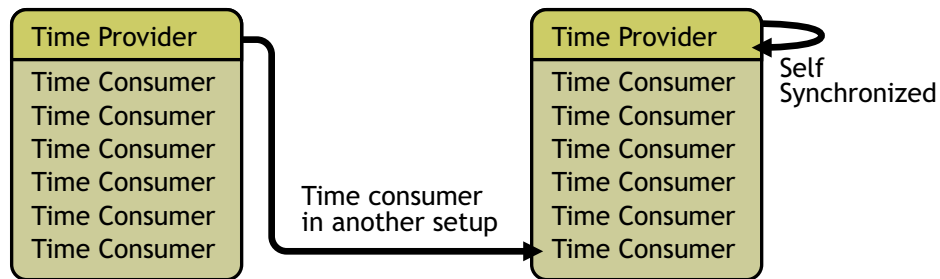
To do this, enter the following at the command prompt:

```
Load XNTPD
```

For more information, see [“Client-Server Mode” on page 13](#).

### Self-Synchronized Mode

In this mode, the server takes time from its own local clock as shown in the figure below.

**Figure 7 Time Group with Self-Synchronization**

To configure the time provider:

- 1** Add lines similar to the following to the time provider's `ntp.conf` file:

```
server 127.127.1.0 prefer
fudge 127.127.1.0 stratum 3
```

- 2** Load XNTPD for the changes to take effect.

To do this, enter the following at the command prompt:

```
Load XNTPD
```

## Configuring the Time Consumers

- 1** Add a line similar to the following to the time consumer's `ntp.conf` file:

```
server IP_address_of_time_provider prefer
```

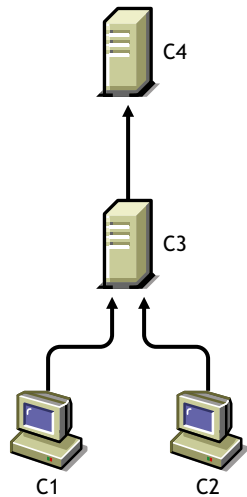
- 2** Load XNTPD for the changes to take effect.

To do this, enter the following at command prompt:

```
Load XNTPD
```

## Sample Scenario

This sample scenario demonstrates how to configure a time-synchronized setup in the manual mode. Consider the scenario in [Figure 8 on page 20](#).

**Figure 8 Sample Scenario for Manual Configuration**

In this scenario:

- ♦ C1 and C2 are time consumers that obtain time from the time provider C3.
- ♦ C3 is also a time consumer that obtains time from the time provider C4.
- ♦ C4 is self-synchronized.

To configure the setup explained in the scenario using manual configuration:

- 1** In the `ntp.conf` files of C1 and C2, add a line similar to the following:

```
server IP_address_of_C3 prefer
```

- 2** In the `ntp.conf` file of C3, add a line similar to the following:

```
server IP_address_of_C4 prefer
```

- 3** In the `ntp.conf` file of C4, add a line similar to the following:

```
server 127.127.1.0 prefer
```

```
fudge 127.127.1.0 stratum 3
```

- 4** Load XNTPD for the changes to take effect.

To do this, enter the following at the command prompt:

```
Load XNTPD
```

## Auto Configuration

Auto configuration is best suited for setups where the time provider is expected to change frequently because it does not require the time consumer to be reconfigured when the time provider changes.

Thus, the setup can be reconfigured in a single step without modifying the configuration of the time consumers in the network.

You can configure a time-synchronized setup using auto configuration by completing the following tasks:

1. [Planning the Setup.](#)
2. [Configuring the Time Providers.](#)
3. [Configuring the Time Consumers.](#)

## Planning the Setup

You must plan the setup to identify a time provider. Identify the most reliable server in the subnet and make it the broadcast or multicast server.

## Configuring the Time Provider

### Broadcast

Add the following line in the time provider's `ntp.conf` file (located in `sys:\etc`) to broadcast the time synchronization service on the network:

```
broadcast subnet_broadcast_address key key_ID
```

### Multicast

Add the following line in the time provider's `ntp.conf` file (located in `sys:\etc`) to multicast the time synchronization service on the network:

```
broadcast 224.0.1.1 key key_ID
```

## Configuring the Time Consumers

### Broadcast

To make the time consumer listen to the broadcast of the time provider's time synchronization service, add the following line to its `ntp.conf` file.

```
broadcastclient subnet_broadcast_address
```

**NOTE:** The *subnet broadcast address* variable is optional.

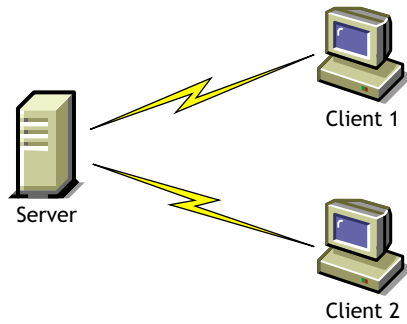
### Multicast

To make the time consumer listen to the multicast of the time provider's time synchronization service, add the following line to its `ntp.conf` file.

```
multicastclient
```

## Sample Scenario

This sample scenario demonstrates how to configure a setup using auto configuration. Consider the scenario in the following figure.

**Figure 9 Sample Scenario for Auto Configuration**

In this scenario:

- ♦ Server1 broadcasts its time synchronization service to all the time consumers in the subnet
- ♦ Client1 and Client2 are two time consumers in the subnet that listen to the broadcast

To configure the setup in this scenario using auto configuration:

- 1** Add the following line to the ntp.conf file of Server1:

```
broadcast subnet_broadcast_address key key_ID
```

- 2** Add the following line to the ntp.conf file of Client1 and Client2:

```
broadcastclient subnet_broadcast_address
```

- 3** Load XNTPD for the changes to take effect.

To do this, enter the following at the command prompt:

```
Load XNTPD
```

## Wide Area Configuration

Wide area configuration is best suited for setups where the servers are spread across geographical locations. It is also suitable for setups which need fault tolerance.

Wide area configuration can be achieved by completing the following tasks:

1. **Planning the Setup.**
2. **Configuring the Time Provider Group.**
3. **Configuring the Time Consumers.**

### Planning the Setup

Create a plan for configuring the time provider group, which is a set of servers configured to ensure fault tolerance and optimal network usage.

### Configuring the Time Provider Group

A time provider group consists of a set of servers that synchronize time in a fault tolerance setup and minimize network traffic.

## Setting Fault Tolerance

- 1 Configure at least two servers to communicate with each other in the peer-to-peer mode by adding a line similar to the following to each server's ntp.conf file:

```
peer IP_address_of_peer
```

- 2 Configure the servers to contact their own unique external time sources in the client-server mode by adding a line similar to the following to each server's ntp.conf file:

```
server IP_address_of_own_external_time_source
```

If one external time source link goes down, both time providers would not lose time synchronization.

- 3 Configure the servers to fall back to their local clocks (self-synchronize) and ensure that the external time source gets preference over the local clock. To achieve this, use a lower preference for the local clock (stratum value of 3). Add lines similar to the following to each server's ntp.conf file:

```
server 127.127.1.0
```

```
fudge 127.127.1.0 stratum 3
```

## Minimizing Network Traffic

To minimize network traffic, configure two servers, on either side of the network, in either the peer-to-peer mode or the client-server mode. These two servers can either have their own external sources or be self-synchronized.

For more information, see [“Peer-to-Peer Mode” on page 14](#) and [“Client-Server Mode” on page 13](#).

## Configuring the Time Consumers

### Setting Fault Tolerance

To set fault tolerance, configure all the time consumers to have at least two time providers either in the client-server mode or the broadcast/multicast mode.

For more information, see [“Client-Server Mode” on page 13](#) and [“Broadcast and Multicast Mode” on page 14](#).

### Minimizing Network Traffic

To minimize network traffic, time consumers should not contact the time providers that are across costly WANs. Preferably, a time consumer should contact a time provider within its own local network.

You can use either manual configuration or auto configuration to configure a time consumer.

To use manual configuration, add lines similar to the following to each time consumer's ntp.conf file:

```
server IP_address_of_time_provider1_within_same_network
```

```
server IP_address_of_time_provider2_within_same_network
```

or

To use auto configuration, add lines similar to the following to each time consumer's ntp.conf file:

```
broadcastclient subnet_broadcast_address
```

or

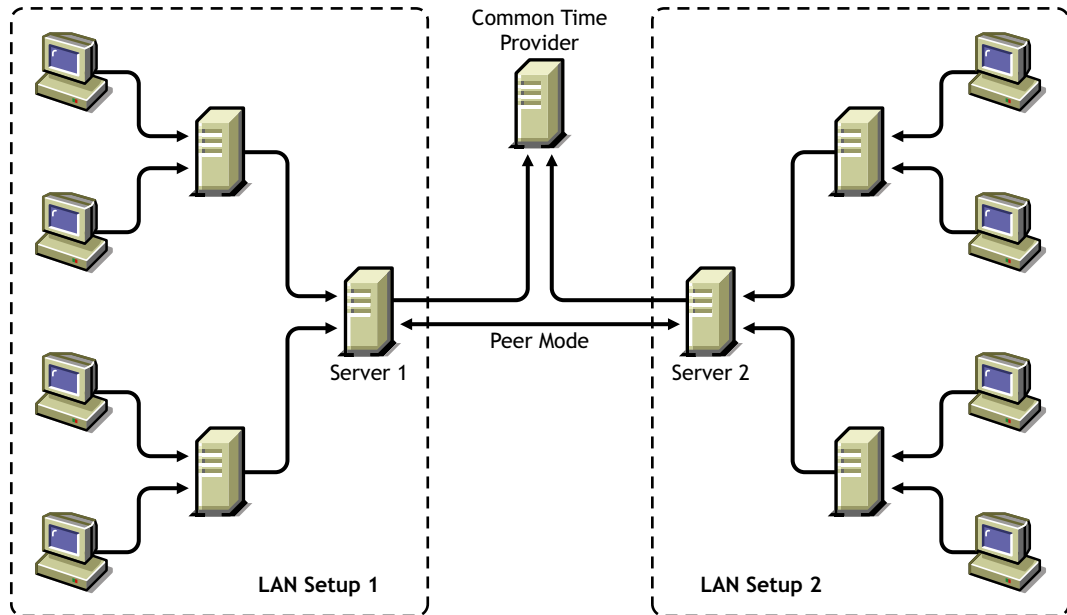
`multicastclient`

## Sample Scenario

This sample scenario demonstrates how to configure a setup using wide area configuration.

Consider the scenario in the following figure.

**Figure 10 Sample Scenario for Wide Area Configuration**



In this scenario,

- ♦ There are two LAN setups. The time consumers and time providers in both the setups are synchronized for time.

Refer to [“Manual Configuration” on page 17](#) and [“Auto Configuration” on page 20](#) for more information.

- ♦ In both LAN setups, only Server1 and Server2 obtain time from sources outside the local network.
- ♦ Server1 and Server2 are time consumers that
  - ♦ Firstly, obtain time from a common time provider in the client-server mode.
  - ♦ Secondly, obtain time from each other in the peer mode.
  - ♦ Thirdly, fall back to their local clock if the external time sources fail.

To configure the setup in this scenario using auto configuration:

- 1 Configure Server1 to obtain time from the common time provider in the client-server mode and make this the preferred time provider by adding a line similar to the following to Server1's `ntp.conf` file:

```
server IP_address_of_Common_Time_Provider prefer
```



- 2** Configure Server1 to obtain time from its peer, Server2 in the peer mode.

```
peer IP_address_of_Server2
```

- 3** Configure Server1 to fall back to its own local clock. Set a low stratum value for the local clock so that preference is given to the external time sources. By setting a low stratum value, it will fall back to its local clock only when the other sources fail.

```
server 127.127.1.0
```

```
fudge 127.127.1.0 stratum 3
```

- 4** Load XNTPD for the changes to take effect by entering the following at command prompt:

```
Load XNTPD
```

- 5** Repeat this procedure, with the appropriate changes to Server2's ntp.conf file, to configure Server2.

You are minimizing network traffic by configuring only Server1 and Server2 to obtain time from a time provider outside the local network.

## Synopsis

The following table summarizes which lines should to be added to the ntp.conf file in each scenario. This file is located in sys:\etc\ntp.conf.

**Table 1**    **Synopsis**

Configuration Type	Time Provider	Time Consumer	Best-Suited Scenario
Manual configuration for client-server mode	server IP_address_of_time_provider prefer	server IP_address_of_time_provider prefer	<ul style="list-style-type: none"> <li>Fewer than 15 servers</li> <li>In the same geographical location</li> </ul>
Manual configuration for self-synchronization	server 127.127.1.0 prefer  fudge 127.127.1.0 stratum 3	N/A	N/A
Auto configuration for broadcast	broadcast subnet_broadcast_address key key_ID	broadcastclient subnet_broadcast_address	<ul style="list-style-type: none"> <li>Time provider changes frequently</li> <li>Time provider is unknown</li> </ul>
Auto configuration for multicast	broadcast 224.0.1.1 key key_ID	multicastclient	<ul style="list-style-type: none"> <li>Time provider changes frequently</li> <li>Time provider is unknown</li> <li>Generates lesser traffic when compared to broadcast</li> </ul>

Configuration Type	Time Provider	Time Consumer	Best-Suited Scenario
Wide area configuration for peer mode or a combination of more than one mode with fault tolerance	server <i>IP_address_of_exte rnal_time_source</i>	Either manual or auto type can be used:  server <i>IP_address_of_tim e_provider1_withi n_same_network</i>	♦ Servers span across geographical locations.
	peer <i>IP_address_of_peer</i>	<i>e_provider1_withi n_same_network</i>	♦ Servers are synchronized in a combination of more than one mode.
	server 127.127.1.0 fudge 127.127.1.0 stratum 3	server <i>IP_address_of_tim e_provider_2_with in_same_network</i>	♦ Need for fault tolerance and minimizing network traffic.
		OR  broadcastclient <i>subnet_broadcast_ address</i>	

## Remote Configuration

NTP can be configured remotely from Novell® Remote Manager (NORM).

NORM displays the configuration file (ntp.conf) of all the servers in the eDirectory™ 8.7.3 tree that you have authenticated to.

You have the following options after making changes to the ntp.conf file:

- ♦ **Save** saves the configuration file.
- ♦ **Apply** saves the configuration file and restarts XNTPD to reflect the new changes.
 

**NOTE:** If a tree is configured with multiple servers running NTP and Timesync, pressing Apply for the server running Timesync, will unload Timesync and load XNTPD.
- ♦ **Restart** restarts XNTPD without saving the changes.

# 4

## Migrating TimeSync Servers to NTP

You can now use the iManager Web-based administration to migrate servers from TimeSync to NTP.

To migrate the servers, perform the following tasks in the order listed below:

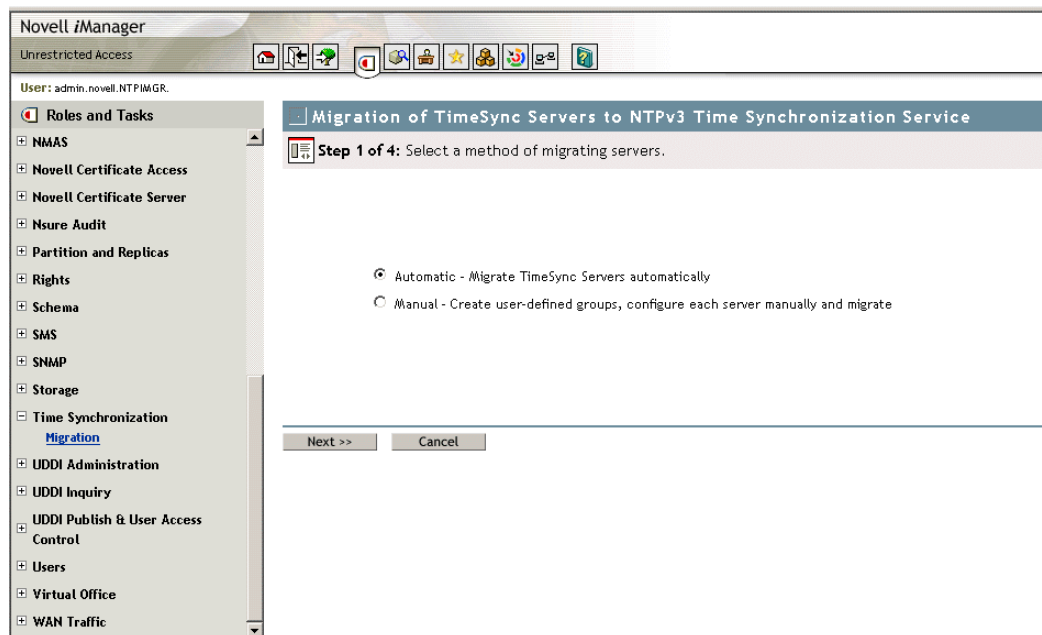
- ♦ “Reviewing Pre-requisites” on page 27
- ♦ “Selecting the Mode of Migration” on page 27
- ♦ “Configuring Servers Running the NTP Services” on page 30
- ♦ “Monitoring the Health Status of Migrated Servers” on page 32

### Reviewing Pre-requisites

- ☐ Ensure that the NTP iManager plug-in (ntptimesync.npm) is located in the root directory of the product. For more information see, Downloading and Installing iManager Plug-in in the *Novell iManager 2.5 Administration Guide* ([http://www.novell.com/documentation/imanager25/index.html?page=/documentation/imanager25/imanager\\_admin\\_25/data/hk42s9ot.html#bktitle](http://www.novell.com/documentation/imanager25/index.html?page=/documentation/imanager25/imanager_admin_25/data/hk42s9ot.html#bktitle))
- ☐ The servers to be migrated must be Netware® 6.5 servers existing on the same Novell® eDirectory™ 8.7.3 tree and must be running on TimeSync.

### Selecting the Mode of Migration

- 1 In iManager, click Time Synchronization > Migration to display the Migrate Time Servers page.

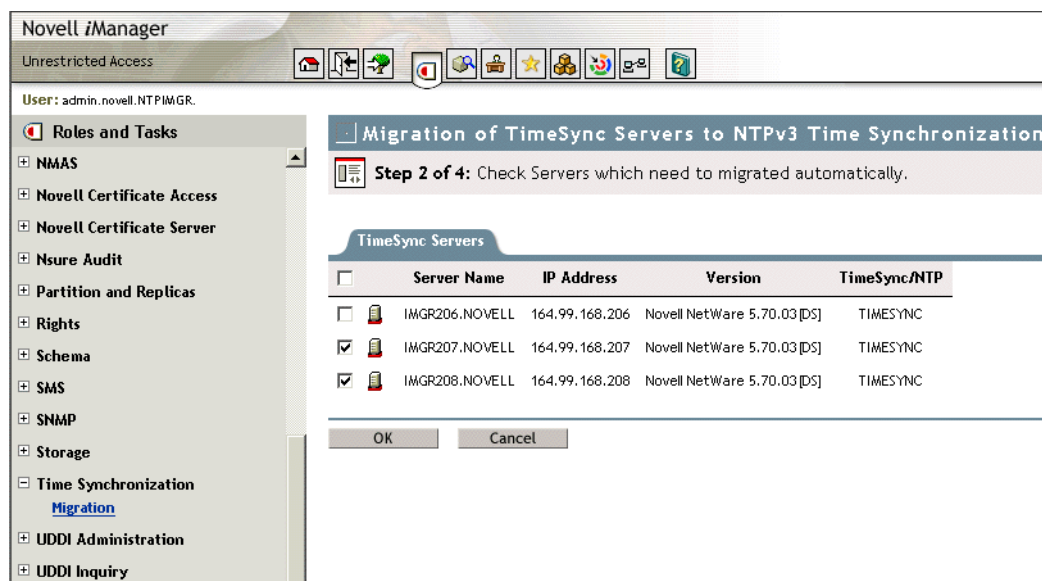


- 2 Select the method of migration, Automatic or Manual.

## Automatically Migrating Servers

This is the default migration method and it can be used by a user who is new to NTP and prefers time synchronization services to automatically set up in the eDirectory tree. The servers are migrated to NTPv3 based on a one-to-one mapping of timesync.cfg to ntp.conf.

- 1 Select the servers to be migrated.



- 2 Click OK to confirm the migration.

A time-out screen is displayed for some time, after which the status of monitored servers is displayed.

For more information on monitoring the status of migrated servers, see [“Monitoring the Health Status of Migrated Servers”](#) on page 32.

## Manually Migrating Servers

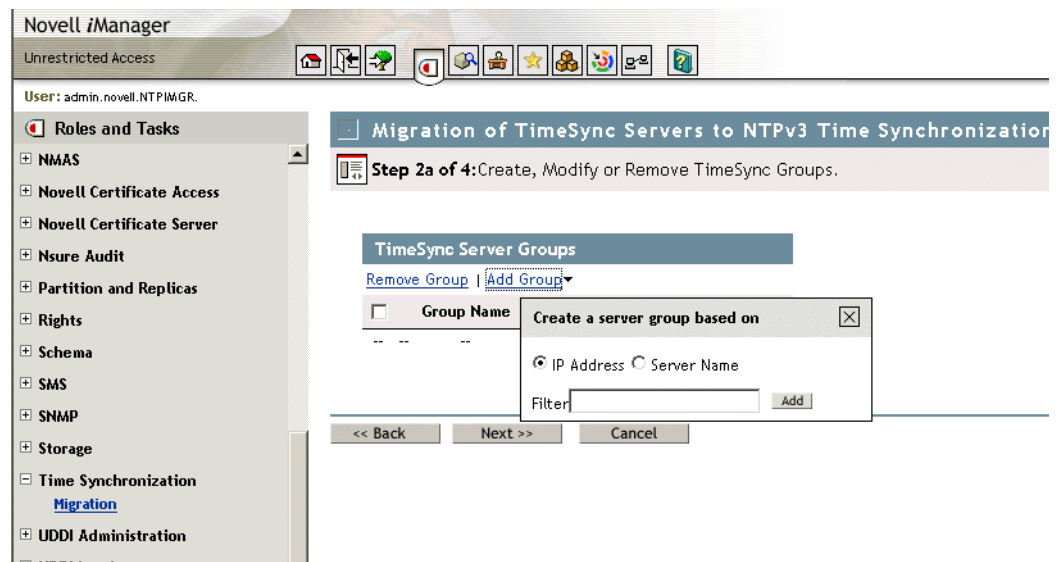
This method can be used by a user who is familiar with NTP. It helps you create a group of servers and configure individual servers. The manual migration process comprises the following tasks:

- ♦ [Adding or Removing TimeSync Groups \(page 29\)](#)
- ♦ [Adding or Removing Servers from a Group \(page 29\)](#)
- ♦ [Configuring Servers Running the NTP Services \(page 30\)](#)

### Adding or Removing TimeSync Groups

#### To add a TimeSync group

- 1 On the TimeSync Server Group page, click Add Group.



- 2 Select the servers to be added to the group, based on the IP address or server name filter, then click Next.

**IMPORTANT:** The only permitted wildcard filter is the asterisk (\*)

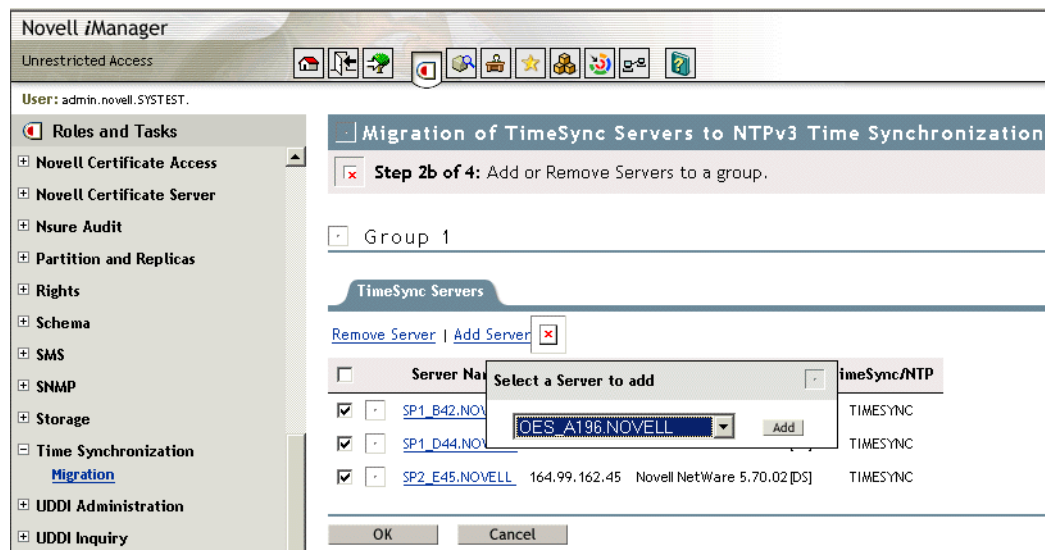
#### To remove a TimeSync Group

On the TimeSync Server Group page, select the group to be excluded and click Remove Group.

### Adding or Removing Servers from a Group

#### To add a server to a group

- 1 Click Add Server to display a drop-down list.



- 2 Select the server you want, then click Add.

#### To remove a server from group

Select one or more servers from the list displayed, then click Remove Server.

## Configuring Servers Running the NTP Services

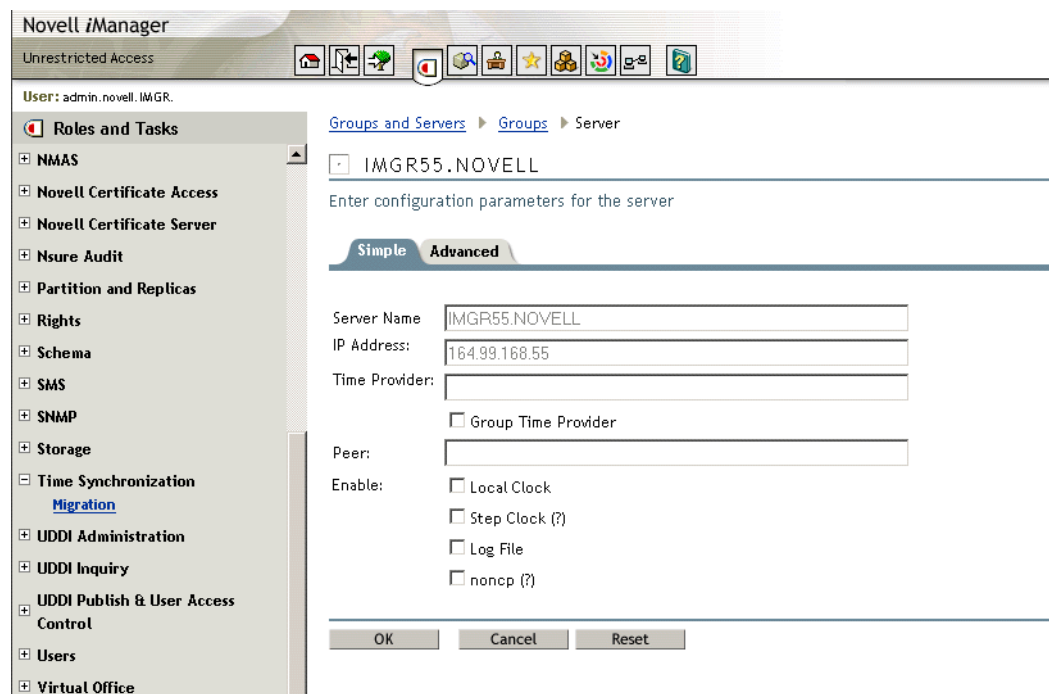
In an automatic migration, all the configuration for migrating the servers is done through backend process.

During a manual migration, you can configure the parameters of each server in a group through a simple or advanced configuration process.

#### Simple Configuration

Each server in a particular group can be configured individually on certain selected NTP parameters.

- 1 On the Configuration Parameters page, specify values for Time Provider and Peer.
- 2 Specify the options to enable on the time provider.



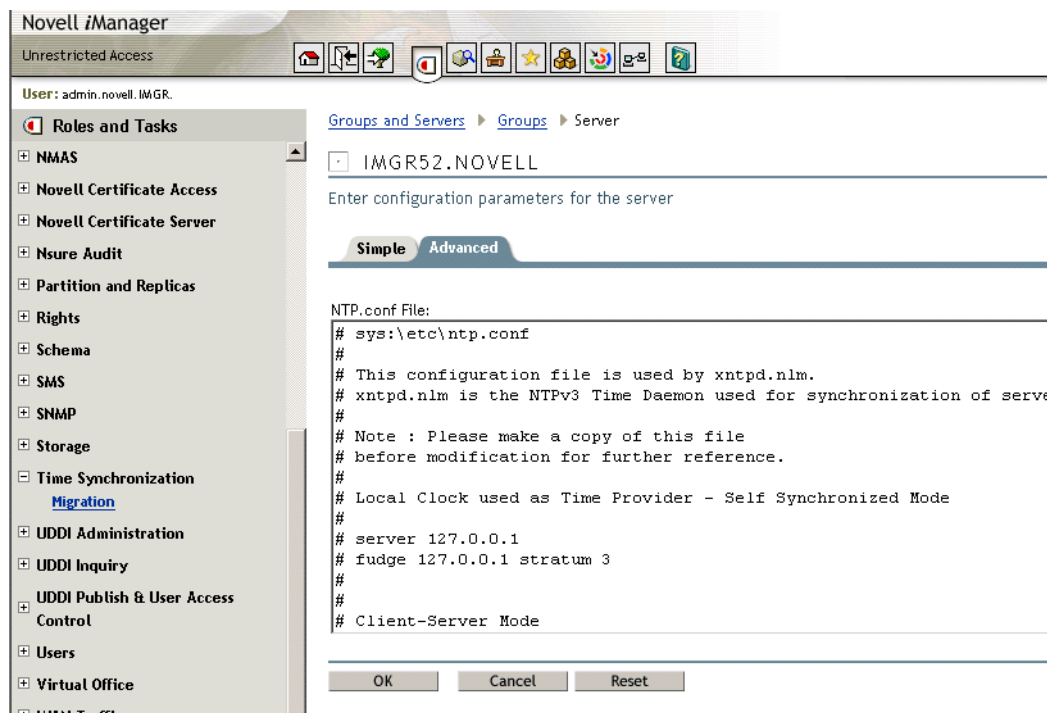
**3** Click OK to update the configuration file of the server.

**4** Click Reset to apply the changes made.

### Advanced Configuration

This mode of editing is recommended for users familiar with NTP. It helps you to edit the `ntp.conf` file for making configuration changes.

**1** After configuring the server parameters, Click Ok to update them.

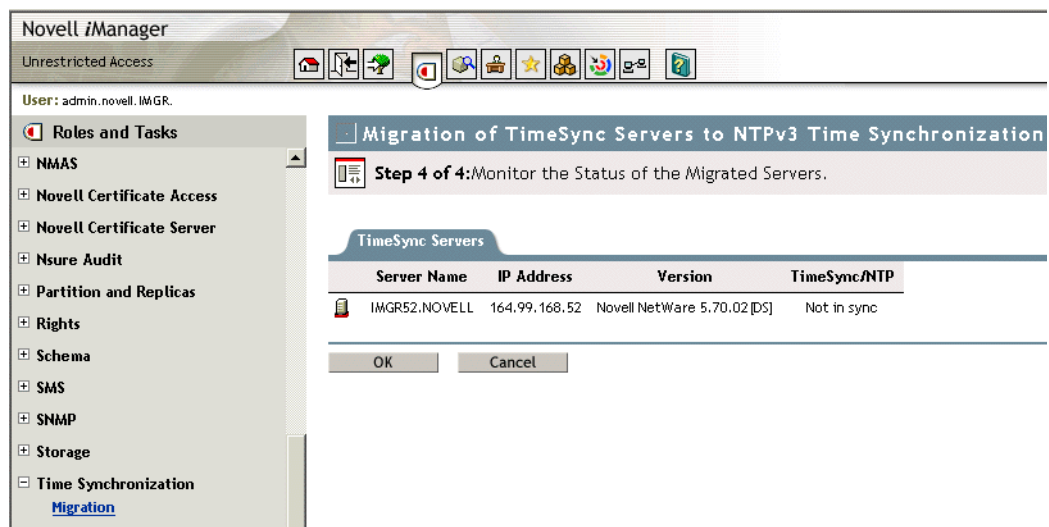


## Monitoring the Health Status of Migrated Servers

The Monitor Configured Groups page lets you view the list of servers which have been migrated.

The successful servers are the ones which are correctly configured and for which the configurations are properly updated. The unsuccessful servers for each group consist of servers for which the configurations are not applied for the particular remote server.

- 1 Click OK to end the migration task.



This screen enables you to detect whether xntpd.nlm is successfully loaded or not.



# 5

## NTP Utilities

This chapter explains the following NTP utilities necessary to synchronize time:

- ♦ [NTPDate \(page 33\)](#)
- ♦ [NTPQ \(page 34\)](#)
- ♦ [NTPTrace \(page 38\)](#)
- ♦ [XNTPDC \(page 39\)](#)
- ♦ [XNTPD \(page 40\)](#)

### NTPDate

NTPDate is used to set the local time and date using NTP servers. The NTP servers are determined by specifying them in the NTPDate command line argument.

Usage:

```
NTPDate [-bBdhoquv] [-a key] [-e authdelay] [-k keyfile] [-o
version] [-p samples] [-s logfile] [-t timeout] server [ ... ]
```

**Table 2** NTPDate Parameters

Parameter	Description
-a <i>key</i>	Enables the authentication function and specifies the key identifier to be used for authentication as the argument keyntptime. The keys and key identifiers must match in both the client and server key files.  The default is to disable the authentication function.
-B	Forces the time to always be slewed, even if the measured offset is greater than +-128 ms.  The default is to step the time if the offset is greater than +-128 ms.  <b>NOTE:</b> If the offset is greater than +-128 ms, it takes a long time (hours) to slew the clock to the correct value. During this time, the host should not be used to synchronize clients.
-b	Forces the time to be stepped, rather than slewed (default). This option should be used when called from a startup file at boot time.
-d	Enables the debugging mode. In the debugging mode, NTPDate goes through all the steps, but does not adjust the local clock. Information useful for general debugging is also printed.

Parameter	Description
-e <i>authdelay</i>	Specifies the processing delay to perform an authentication function as the value <i>authdelay</i> , in seconds and fraction (see “XNTPD” on page 40 for details). This number is usually small enough to be negligible for most purposes, though specifying a value might improve timekeeping on very slow CPUs.
-h	Displays the help.
-k <i>keyfile</i>	Specifies the path for the authentication key file as the string <i>keyfile</i> . The default is <code>sys:\etc\ntp.keys</code> . This file should be in the format described in XNTPD.
-o <i>version</i>	Specifies the NTP version for outgoing packets as the integer <i>version</i> , which can be 1 or 2. The default is 3. This allows NTPDate to be used with older NTP versions.
-p <i>samples</i>	Specifies the number of samples to be acquired from each server as the integer <i>samples</i> , with values from 1 to 8 inclusive. The default is 4.
-q	Query only. Do not set the clock.
-s <i>logfile</i>	Enables logging of the XNTPD progress screen into the <i>logfile</i> .
-t <i>timeout</i>	Specifies the maximum time waiting for a server response as the value <i>timeout</i> , in seconds and fraction. The value is rounded off to a multiple of 0.2 seconds. The default is 1 second, a value suitable for polling across a LAN.
-u	Directs NTPDate to use an unprivileged port or outgoing packets. This is most useful when you are behind a firewall that blocks incoming traffic to privileged ports and you want to synchronize with hosts beyond the firewall. Note that the -d option always uses unprivileged ports.
-v	Be verbose. This option logs NTPDate's version identification string.

## NTPQ

NTPQ is an interactive NTPv3 query utility to help query the status or quality of time parameters.

Usage:

```
NTPQ [-i np] [-c command] [host] [ ... ]
```

**Table 3** NTPQ Parameters

Parameter	Description
-c	Is interpreted as an interactive format command and is added to the list of commands to be executed on the specified hosts. Multiple -c options can be given.
-i	Forces NTPQ to operate in the interactive mode. Prompts will be written to the standard output and commands read from the standard input.
-n	Output all host addresses in dotted-quad numeric format rather than converting to the canonical hostnames.
-p	Prints a list of the peers known to the server as well as a summary of their state.

## Internal Commands

Interactive format commands consist of a keyword followed by zero to four arguments. Only enough characters of the full keyword to uniquely identify the command need be typed. The output of a command is normally sent to the standard output, but optionally the output of individual commands may be sent to a file by appending a "<", followed by a file name, to the command line. A number of interactive format commands are executed entirely within the NTPQ program itself and do not result in NTP mode 6 requests being sent to a server. These are described following.

**Table 4**     **Parameter Description**

Parameter	Description
? [ <i>command_keyword</i> ]	A "?" by itself will print a list of all the command keywords known to this incarnation of NTPQ. A "?" followed by a command keyword will print function and usage information about the command.
help [ <i>command_keyword</i> ]	
addvars <i>variable_name</i> [ = <i>value</i> ] [ ... ]	The data carried by NTP mode 6 messages consists of a list of items of the form <i>variable_name</i> = <i>value</i> , where the " = <i>value</i> " is ignored, and can be omitted, in requests to the server to read variables. NTPQ maintains an internal list in which data to be included in control messages can be assembled, and sent using the readlist and writelist commands described below. The addvars command allows variables and their optional values to be added to the list. If more than one variable is to be added, the list should be comma-separated and not contain white space. The rmvars command can be used to remove individual variables from the list, while the clearlist command removes all variables from the list.
rmvars <i>variable_name</i> [ ... ]	
clearvars	
authenticate yes   no	Normally NTPQ does not authenticate requests unless they are write requests. The command authenticate yes causes NTPQ to send authentication with all requests it makes. Authenticated requests causes some servers to handle requests slightly differently, and can occasionally melt the CPU in fuzzballs if you turn authentication on before doing a peer display.
cooked	Causes output from query commands to be "cooked". Variables which are recognized by the server will have their values reformatted for human consumption. Variables which NTPQ thinks should have a decodeable value but didn't are marked with a trailing "?".
debug more   less   off	Turns internal query program debugging on and off.
delay <i>milliseconds</i>	Specify a time interval to be added to timestamps included in requests which require authentication. This is used to enable (unreliable) server reconfiguration over long delay network paths or between machines whose clocks are unsynchronized. Actually the server does not now require timestamps in authenticated requests, so this command may be obsolete.
host <i>hostname</i>	Set the host to which future queries will be sent. Hostname may be either a host name or a numeric address.
hostnames [ yes   no ]	If "yes" is specified, host names are printed in information displays. If "no" is specified, numeric addresses are printed instead. The default is "yes", unless modified using the command line -n switch.
keyid <i>keyid</i>	This command allows the specification of a key number to be used to authenticate configuration requests. This must correspond to a key number the server has been configured to use for this purpose.
ntpversion 1   2   3	Sets the NTP version number which NTPQ claims in packets. Defaults to 3, Note that mode 6 control messages (and modes, for that matter) didn't exist in NTP version 1. There appear to be no servers left which demand version 1.

Parameter	Description
quit	Exit NTPQ.
passwd	This command prompts you to type in a password (which will not be echoed) which will be used to authenticate configuration requests. The password must correspond to the key configured for use by the NTP server for this purpose if such requests are to be successful.
raw	Causes all output from query commands is printed as received from the remote server. The only formatting/interpretation done on the data is to transform nonascii data into a printable (but barely understandable) form.
timeout <i>milliseconds</i>	Specify a timeout period for responses to server queries. The default is about 5000 milliseconds. Note that since NTPQ retries each query once after a timeout, the total waiting time for a timeout will be twice the timeout value set.

## Control Message Commands

Each peer known to an NTP server has a 16 bit integer association identifier assigned to it. NTP control messages which carry peer variables must identify the peer the values correspond to by including its association ID. An association ID of 0 is special, and indicates the variables are system variables, whose names are drawn from a separate name space.

Control message commands result in one or more NTP mode 6 messages being sent to the server, and cause the data returned to be printed in some format. Most commands currently implemented send a single message and expect a single response. The current exceptions are the peers command, which will send a preprogrammed series of messages to obtain the data it needs, and the mreadlist and mreadvar commands, which will iterate over a range of associations.

**Table 5**      **Parameter Description**

Parameter	Description
associations	Obtains and prints a list of association identifiers and peer statuses for in-spec peers of the server being queried. The list is printed in columns. The first of these is an index numbering the associations from 1 for internal use, the second the actual association identifier returned by the server and the third the status word for the peer. This is followed by a number of columns containing data decoded from the status word. Note that the data returned by the "associations" command is cached internally in NTPQ. The index is then of use when dealing with stupid servers which use association identifiers which are hard for humans to type, in that for any subsequent commands which require an association identifier as an argument, the form &index may be used as an alternative.
clockvar [ <i>assocID</i> ] [ <i>variable_name</i> [ = cv [ <i>assocID</i> ] [ <i>variable_name</i> [ = value [ ... ] ] [ ... ]	Requests that a list of the server's clock variables be sent. Servers which have a radio clock or other external synchronization will respond positively to this. If the association identifier is omitted or zero the request is for the variables of the "system clock" and will generally get a positive response from all servers with a clock. If the server treats clocks as pseudo-peers, and hence can possibly have more than one clock connected at once, referencing the appropriate peer association ID will show the variables of a particular clock. Omitting the variable list will cause the server to return a default variable display.

Parameter	Description
lassociations	Obtains and prints a list of association identifiers and peer statuses for all associations for which the server is maintaining state. This command differs from the "associations" command only for servers which retain state for out-of-spec client associations (i.e., fuzzballs). Such associations are normally omitted from the display when the "associations" command is used, but are included in the output of "lassociations".
lpassociations	Print data for all associations, including out-of-spec client associations, from the internally cached list of associations. This command differs from "passociations" only when dealing with fuzzballs.
lpeers	Like R peers, except a summary of all associations for which the server is maintaining state is printed. This can produce a much longer list of peers from fuzzball servers.
mreadlist <i>assocID assocID</i> mrl <i>assocID assocID</i>	Like the readlist command, except the query is done for each of a range of (nonzero) association IDs. This range is determined from the association list cached by the most recent associations command.
mreadvar <i>assocID assocID</i> [ <i>variable_name</i> [ = <i>value</i> [ ... ] ] mrv <i>assocID assocID</i> [ <i>variable_name</i> [ = <i>value</i> [ ... ] ]	Like the readvar command, except the query is done for each of a range of (nonzero) association IDs. This range is determined from the association list cached by the most recent associations command.
opeers	An old form of the peers command with the reference ID replaced by the local interface address.
passociations	Prints association data concerning in-spec peers from the internally cached list of associations. This command performs identically to the "associations" except that it displays the internally stored data rather than making a new query.
peers	<p>Obtains a list of in-spec peers of the server, along with a summary of each peer's state. Summary information includes the address of the remote peer, the reference ID (0.0.0.0 if the refID is unknown), the stratum of the remote peer, the type of the peer (local, unicast, multicast or broadcast), when the last packet was received, the polling interval, in seconds, the reachability register, in octal, and the current estimated delay, offset and dispersion of the peer, all in seconds.</p> <p>The character in the left margin indicates the fate of this peer in the clock selection process. The codes mean: &lt;sp&gt; discarded due to high stratum and/or failed sanity checks; "x" designated falsticker by the intersection algorithm; "." culled from the end of the candidate list; "-" discarded by the clustering algorithm; "+" included in the final selection set; "#" selected for synchronization but distance exceeds maximum; "*" selected for synchronization; and "o" selected for synchronization, PPS signal in use.</p> <p><b>NOTE:</b> Since the peers command depends on the ability to parse the values in the responses it gets it may fail to work from time to time with servers which poorly control the data formats.</p> <p>The contents of the host field may be one of four forms. It may be a host name, an IP address, a reference clock implementation name with its parameter or "REFCLK(&lt;implementation number&gt;, &lt;parameter&gt;)". On "hostnames no" only IP- addresses will be displayed.</p>

Parameter	Description
<code>pstatus assocID</code>	Sends a read status request to the server for the given association. The names and values of the peer variables returned will be printed. Note that the status word from the header is displayed preceding the variables, both in hexadecimal and in pidgeon English.
<code>readlist [ assocID ]</code> <code>rl [ assocID ]</code>	Requests that the values of the variables in the internal variable list be returned by the server. If the association ID is omitted or is 0 the variables are assumed to be system variables. Otherwise they are treated as peer variables. If the internal variable list is empty a request is sent without data, which should induce the remote server to return a default display.
<code>readvar assocID variable_name [ = value ]</code> [ ... ] <code>rv assocID [ variable_name [ = value ] [ ... ]</code>	Requests that the values of the specified variables be returned by the server by sending a read variables request. If the association ID is omitted or is given as zero the variables are system variables, otherwise they are peer variables and the values returned will be those of the corresponding peer. Omitting the variable list will send a request with no data which should induce the server to return a default display.
<code>rvi index</code>	Similar to <code>rv</code> with association ID corresponding to this index.
<code>writevar assocID variable_name [ = value ]</code> [ ... ]	Similar to <code>readvar</code> request, except the specified variables are written instead of read.
<code>writelist [ assocID ]</code>	Similar to <code>readlist</code> request, except the internal list variables are written instead of read.
<code>showipconf</code>	Displays the IP addresses of the local machine along with the corresponding subnet broadcast addresses.

## NTPTrace

NTPTrace is a utility to query the time server and its servers until the master server is queried. NTPTrace determines where a given NTP server gets its time from, and follows the chain of NTP servers back to their master time source. If given no arguments, it starts with local host. An example of the output from NTPTrace is given below:

```
% ntptrace
localhost: stratum 4, offset 0.0019529, synch distance 0.144135
server2ozo.com: stratum 2, offset 0.0124263, synch distance 0.115784
usndh.edu: stratum 1, offset 0.0019298, synch distance 0.011993, refid
'WWVB'
```

On each line, the fields are (left to right):

- ◆ hostname
- ◆ host stratum; stratum is the server hop count to the primary source.
- ◆ time offset between that host and the local host (as measured by NTPTrace; this is why it is not always zero for "local host"). The time unit is given in seconds.
- ◆ host synchronization distance; synchronization distance is the estimated error relative to the primary source
- ◆ reference clock ID (only for stratum-1 servers)

Usage:

```
NTPTrace [ -dhnv ] [ -r retries ] [ -t timeout ] [ server ]
```

**Table 6** NTPTrace Parameters

Parameter	Description
-d	Turns on some debugging output.
-h	Displays the help.
-n	Turns off the printing of hostnames; instead, host IP addresses are given. This can be useful if a nameserver is down.
-r <i>retries</i>	Sets the number of retransmission attempts for each host (default = 5).
-t <i>timeout</i>	Sets the retransmission timeout (in seconds) (default = 2).
-v	Prints verbose information about the NTP servers.

## XNTPDC

XNTPDC is the remote configuration utility. It is used to query the XNTPD daemon about its current state and to request changes in that state.

If one or more request options are included on the command line when XNTPDC is executed, each of the requests is sent to the NTP servers running on each of the hosts given as command line arguments, or on localhost by default. If no request options are given, XNTPDC attempts to read commands from the standard input and executes these on the NTP server running on the first host given on the command line, again defaulting to localhost when no other host is specified. XNTPDC prompts for commands if the standard input is a terminal device.

The operations of XNTPDC are specific to the particular implementation of the XNTPDC daemon and can be expected to work only with this and maybe some previous versions of the daemon. Requests from a remote XNTPDC program that affect the state of the local server must be authenticated, which requires both the remote program and local server share a common key and key identifier.

```
XNTPDC [ -i lnps ] [ -c command ] [ host ] [ ... ]
```

Specifying a command line option other than **-i** or **-n** causes the specified query (queries) to be sent to the indicated hosts immediately. Otherwise, XNTPDC will attempt to read interactive format commands from the standard input.

**Table 7** XNTPDC Parameters

Parameter	Description
-c <i>command</i>	The following argument is interpreted as an interactive format command and is added to the list of commands to be executed on the specified hosts. Multiple -c options can be given.
-i	Forces XNTPDC to operate in interactive mode. Prompts will be written to the standard output and commands read from the standard input.

Parameter	Description
-l	Obtains a list of peers known to the servers. This switch is equivalent to <b>-c listpeers</b> .
-n	Outputs all host addresses in dotted-quad numeric format rather than converting to the canonical hostnames.
-p	Prints a list of the peers known to the server as well as a summary of their state. This is equivalent to <b>-c peers</b> .
-s	Prints a list of the peers known to the server as well as a summary of their state, but in a slightly different format than the -p switch. This is equivalent to <b>-c dmpeers</b> .

## XNTPD

XNTPD is an operating system daemon which sets and maintains the system time-of-day in synchronism with Internet standard time servers.

The daemon can operate in any of several modes, including client-server and broadcast/multicast mode, as described in RFC-1305. A broadcast/multicast client can discover remote servers, compute client-server propagation delay correction factors and configure itself automatically. This makes it possible to deploy a fleet of workstations without specifying configuration details specific to the local environment.

Ordinarily, XNTPD reads the `ntp.conf` configuration file at startup in order to determine the synchronization sources and operating modes. It is also possible to specify a working, although limited, configuration entirely on the command line, obviating the need for a configuration file. This might be particularly appropriate when the local host is to be configured as a broadcast or multicast client, with all peers being determined by listening to broadcasts at run time.

Various internal XNTPD variables can be displayed and configuration options altered while the daemon is running using the NTPQ and XNTPDC utility programs.

Usage:

```
XNTPD [ -aAbdhm ] [ -c configfile ] [ -f driftfile ] [ -k keyfile
] [ -l logfile ] [ -p pidfile ] [ -r broadcastdelay ] [ -s statsdir
] [ -t trustkey ] [ -v variable ] [ -V defaultvariable ] [-T noncp/
slp ] [-S]
```

Table 8 XNTPD Parameters

Parameter	Description
-a	Enables authentication mode. The default is enabled, so this option is obsolete now.
-A	Disables authentication mode.
-b	Synchronizes using NTP broadcast messages.
-c <i>configfile</i>	Specifies the name and path of the configuration file.  <b>NOTE:</b> Novell Remote Manager does not understand this user-defined configuration file, and it opens the default <code>sys:\etc\ntp.conf</code> file



Parameter	Description
-d	Specifies the debugging mode. This flag might occur multiple times, with each occurrence indicating greater detail of display.
-f <i>driftfile</i>	Specifies the name and path of the drift file.
-h	Displays the help.
-k <i>keyfield</i>	Specifies the name and path of the file containing the NTP authentication keys.
-l <i>logfile</i>	Specifies the name and path of the log file. The default is the system log facility.  <b>NOTE:</b> If the -S option (see “-S” on page 41 for more information) is enabled along with -l option, the NTPDate events are also logged into the log file (ntpdate.log).
-m	Synchronizes using NTP multicast messages on the IP multicast group address 224.0.1.1 (requires multicast kernel).
-p <i>pidfile</i>	Specifies the name and path to record the daemon's process ID.
-r <i>broadcastdelay</i>	Specifies the default propagation delay from the broadcast/multicast server and this computer. This is used only if the delay cannot be computed automatically by the protocol.
-s <i>statsdir</i>	Specifies the directory path for files created by the statistics facility.
-t <i>key</i>	Adds a key number to the trusted key list.
-v <i>variable</i>	Adds a system variable.
-V <i>defaultvariable</i>	Adds a system variable listed by default.
-T <i>noncp</i>	-T provides Timesync migration or backward compatibility options.  Prevents running of the ncp engine on XNTPD, which services all ncp time requests from NetWare 4, Novell clients and dsrepair.
-T <i>slp</i>	Enables NTP to automatically discover SLP advertising Timesync SINGLE server on the network and add the Timesync SINGLE server's IP address in the ntp.conf configuration file as a time provider.  <b>WARNING:</b> Do not use this option in the sys:\system\timeserv.ncf file.
-S	XNTPD steps the clock to the time of the best available server by calling NTPDate with the server list from the NTP configuration file (ntp.conf). This sets the clock status to "nearly in sync", meaning that time is synchronized in as close as 0.5 seconds. This basically helps XNTPD to synchronize fast.

## The Configuration File

The XNTPD configuration file (ntp.conf) is read at initial startup in order to specify the synchronization sources, modes and other related information. It is installed in the sys:\system directory, but could be installed elsewhere (see “-c configfile” on page 40).

### Sample Configuration File

The ntp.conf looks similar to the following:

```
# sys:\etc\ntp.conf
```

```
#  
  
# This configuration file is used by xntpd.nlm.  
# xntpd.nlm is the NTPv3 Time Daemon used for  
# synchronization of servers.  
  
#  
# Note : Please make a copy of  
# this file before modification  
# for further reference.  
  
#  
# Local Clock used as Time Provider - Self Synchronized Mode  
  
#  
# server 127.127.1.0  
# fudge 127.127.1.0 stratum 3  
  
#  
# Client-Server Mode  
# <IP Address> : Time provider IP address  
  
#  
# Time Provider  
# server <IP Address>  
  
#  
# Time Provider  
# server <IP Address>  
  
#  
# Peer-Peer Mode  
# <IP Address> : Peer IP address  
  
#  
# peer <IP Address>  
  
#  
# To Configure this NetWare box to Broadcast the "time service"  
  
#  
# broadcast <Subnet broadcast Address> key <key_id>  
#  
# or  
# broadcast 255.255.255.255 key <key_id>  
  
#  
# To Configure this NetWare box to Multicast the "time service"
```

```
#
# broadcast 224.0.1.1 key <key_id>
#
# To Configure NTP Broadcast Client
#
# broadcastclient
#
# To Configure NTP Multicast Client
#
# multicastclient
#
# Authentication Options
#
# enable auth monitor
# keys sys:\etc\ntp.keys
# trustedkey 0
# requestkey 0
# controlkey 0
#
# Backward Compatibility with Timesync
#
# Switch off the Timesync NCP service
# noncp
# Step the time to the source clock for slewing
# stepclock
#
# Monitoring/Logging Options
#
# driftfile sys:\system\drift.ntp
# statsdir sys:\system\
# logfile sys:\system\ntp.log
# filegen peerstats file peerstat type none enable
# filegen loopstats file loopstat type none enable
# filegen clockstats file clkstat type none enable
```

Configuration commands consist of an initial keyword followed by a list of arguments, some of which may be optional, separated by whitespace. Commands may not be continued over multiple lines. Arguments may be host names, host addresses written in numeric, dotted-quad form, integers, floating point numbers (when specifying times in seconds) and text strings. Optional arguments are delimited by [ ] in the following descriptions, while alternatives are separated by |. The notation [ ... ] means an optional, indefinite repetition of the last item before the [ ... ].

See the following for configuration and control options. While there is a rich set of options available, the only required option is one or more server, peer, or broadcast commands described in [“Configuration Options” on page 44](#).

- ♦ [“Configuration Options” on page 44](#)
- ♦ [“Authentication Options” on page 47](#)
- ♦ [“Monitoring Options” on page 49](#)
- ♦ [“Access Control Options” on page 51](#)
- ♦ [“Miscellaneous Options” on page 52](#)

## Files

sys:\etc\ntp.conf - the default name of the configuration file

sys:\system\ntp.drift - the default name of the drift file

sys:\etc\ntp.keys - the default name of the key file

## Configuration Options

```
peer address [ key key ] [ version version ] [ prefer ] [ minpoll minpoll [
maxpoll maxpoll ]
```

```
server address [ key key ] [ version version ] [ prefer ]
```

```
broadcast address [ key key ] [ version version ] [ ttl ttl ]
```

These three commands specify the time server name or address to be used and the mode in which to operate. The address can be either a DNS name or a IP address in dotted-quad notation. The peer command specifies that the local server is to operate in symmetric active mode with the remote server. In this mode, the local server can be synchronized to the remote server and, in addition, the remote server can be synchronized by the local server. This is useful in a network of servers where, depending on various failure scenarios, either the local or remote server may be the better source of time.

The server command specifies that the local server is to operate in client mode with the specified remote server. In this mode, the local server can be synchronized to the remote server, but the remote server can never be synchronized to the local server.

The broadcast command specifies that the local server is to operate in broadcast mode, where the local server sends periodic broadcast messages to a client population at the broadcast/multicast address specified. Ordinarily, this specification applies only to the local server operating as a sender; for operation as a broadcast client, see the broadcastclient or multicastclient commands below. In this mode, address is usually the broadcast address on (one of) the local network(s) or a multicast address assigned to NTP. The Numbers Czar has assigned the address 224.0.1.1 to NTP; this is presently the only address that should be used.

**NOTE:** The use of multicast features requires a multicast kernel, which is not yet ubiquitous in vendor products.

For more information on the configuration options, see [Table 9, “XNTPD Configuration Options,” on page 45.](#)

**Table 9** XNTPD Configuration Options

Parameter	Description
key <i>key</i>	All packets sent to the address are to include authentication fields encrypted using the specified key identifier, which is an unsigned 32 bit integer. The default is to not include an encryption field.
version <i>version</i>	Specifies the version number to be used for outgoing NTP packets. Versions 1, 2, and 3 are the choices, with version 3 the default.
prefer	Marks the server as preferred. All other things being equal, this host will be chosen for synchronization among a set of correctly operating hosts.
ttl <i>ttl</i>	This option is used only with broadcast mode. It specifies the time- to-live ttl to use on multicast packets. Selection of the proper value, which defaults to 127, is something of a black art and must be coordinated with the network administrator(s).
minpoll <i>minpoll</i>	This option specifies the minimum polling interval for NTP messages, in seconds to the power of two. The allowable range is 4 (16 s to 14 (16384 s) inclusive. The default is 6 (64 s) for all except reference clocks.
maxpoll <i>maxpoll</i>	This option specifies the maximum polling interval for NTP messages, in seconds to the power of two. The allowable range is 4 (16 s to 14 (16384 s) inclusive. The default is 10 (1024 s) for all except reference clocks.
broadcastclient [ <i>address</i> ]	This command directs the local server to listen for broadcast messages at the broadcast address <i>address</i> of the local network. The default address is the subnet address with the host field bits set to ones. Upon hearing a broadcast message for the first time, the local server measures the nominal network delay using a brief client/server exchange with the remote server, then enters the broadcastclient mode, in which it listens for and synchronizes to succeeding broadcast messages. Note that, in order to avoid accidental or malicious disruption in this mode, both the local and remote servers should operate using authentication and the same trusted key and key identifier.
multicastclient [ <i>address</i> ] [ ... ]	This command directs the local server to listen for multicast messages at the group address(es) of the global network. The default address is that assigned by the Numbers Czar to NTP (224.0.1.1). This command operates in the same way as the broadcastclient command, but uses IP multicasting. Support for this command requires a multicast kernel.

Parameter	Description
<code>driftfile <i>driftfile</i></code>	<p>This command specifies the name of the file used to record the frequency offset of the local clock oscillator. If the file exists, it is read at startup in order to set the initial frequency offset and then updated once per hour with the current frequency offset computed by the daemon. If the file does not exist or this command is not given, the initial frequency offset is assumed zero. In this case, it may take some hours for the frequency to stabilize and the residual timing errors to subside.</p> <p>The <code>ntp.drift</code> file format consists of a single line containing a single floating point number, which records the frequency offset measured in parts-per-million (PPM). That the file is updated once per hour by first writing the current drift value into a temporary file and then renaming this file to replace the old version. This implies that XNTPD must have write permission for the directory the drift file is located in, and that file system links, symbolic or otherwise, should probably be avoided.</p>

```
enable auth | bclient | monitor | pll | pps | stats
```

```
disable auth | bclient | monitor | pll | pps | stats
```

Provides a way to enable or disable various server options. Flags not mentioned are unaffected.

**NOTE:** All these flags can be controlled remotely using XNTPDC.

**Table 10** XNTPD Parameters for Enabling and Disabling Server Options

Parameter	Description
<code>auth</code>	Enables the server to synchronize with unconfigured peers only if the peer has been correctly authenticated using a trusted key and key identifier. The default for this flag is enable.
<code>bclient</code>	Enables the server to listen for a message from a broadcast or multicast server, as in the <code>multicastclient</code> command with default address. The default for this flag is disable.
<code>monitor</code>	Enables the monitoring facility. See the XNTPDC program and the <code>monlist</code> command or further information. The default for this flag is enable.
<code>pll</code>	Enables the server to adjust its local clock by means of NTP. If disabled, the local clock free-runs at its intrinsic time and frequency offset. This flag is useful in case the local clock is controlled by some other device or protocol and NTP is used only to provide synchronization to other clients. In this case, the local clock driver is used. The default for this flag is enable.
<code>pps</code>	Enables the pulse-per-second (PPS) signal when frequency and time is disciplined by the precision time kernel modifications. The default for this flag is disable.
<code>stats</code>	Enables the statistics facility. The default for this flag is enable.
<code>tick <i>value</i></code>	If no value for tick can be found from the kernel, use this value. This is the "normalized" value; if your system keeps tick in nanoseconds you must divide your value by 1000. The expected range of the value is between 900 and 11,000 (don't use the comma in the config file).

Parameter	Description
tickadj <i>value</i>	If no value for tickadj can be found in the kernel, use this value. The value must be "normalized"; if your system keeps tickadj in nanoseconds you must divide your value by 1000. The expected range of the value is between 1 and 50.

The XNTPD **-S** and **-T noncp** options can also be added in the configuration file as **stepclock** and **noncp** respectively.

**Table 11** Stepclock and Noncp

Parameter	Description
stepclock	XNTPD steps the clock to the time of the best available server by calling NTPDate with the server list from the NTP configuration file (ntp.conf). This sets the clock status to "nearly in sync", meaning that time is synchronized in as close as 0.5 seconds. This basically helps XNTPD to synchronize fast.
noncp	Prevents running of the ncp engine on XNTPD, which services all ncp time requests from NetWare 4, Novell clients and dsrepair.

## Authentication Options

The NTP standard specifies an extension which provides cryptographic authentication of received NTP packets. This is implemented in XNTPD using the DES or MD5 algorithms to compute a digital signature, or message digest. The specification allows any one of possibly four billion keys, numbered with 32-bit key identifiers, to be used to authenticate an association. The servers involved in an association must agree on the key and key identifier used to authenticate their messages.

Keys and related information are specified in a key file, which should be exchanged and stored using secure procedures beyond the scope of the protocol. There are three classes of keys involved in the current implementation. One class is used for ordinary NTP associations, another for the NTPQ utility program and the third for the XNTPDC utility program.

**Table 12** XNTPD Authentication Command Options

Parameter	Description
keys <i>keyfile</i>	Specifies the file name containing the encryption keys and key identifiers used by XNTPD, NTPQ and XNTPDC when operating in authenticated mode. The format of this file is described later in this document.
trustedkey <i>key</i> [ ... ]	Specifies the encryption key identifiers which are trusted for the purposes of authenticating peers suitable for synchronization. The authentication procedures require that both the local and remote servers share the same key and key identifier for this purpose, although different keys can be used with different servers. The key arguments are 32-bit unsigned integers. Note that NTP key 0 is fixed and globally known. If meaningful authentication is to be performed the 0 key should not be trusted.

Parameter	Description
requestkey key	Specifies the key identifier to use with the XNTPDC program, which uses a proprietary protocol specific to this implementation of XNTPD. This program is useful to diagnose and repair problems that affect XNTPD operation. The key argument to this command is a 32-bit unsigned integer. If no requestkey command is included in the configuration file, or if the keys don't match, such requests will be ignored.
controlkey key	Specifies the key identifier to use with the NTPQ program, which uses the standard protocol defined in RFC-1305. This program is useful to diagnose and repair problems that affect the XNTPD operation. The key argument to this command is a 32-bit unsigned integer. If no requestkey command is included in the configuration file, or if the keys don't match, such requests will be ignored.

### Authentication Key File Format

In the case of DES, the keys are 56 bits long with, depending on type, a parity check on each byte. In the case of MD5, the keys are 64 bits (8 bytes). XNTPD reads its keys from a file specified using the -k command line option or the keys statement in the configuration file. While key number 0 is fixed by the NTP standard (as 56 zero bits) and may not be changed, one or more of the keys numbered 1 through 15 may be arbitrarily set in the keys file.

The key file uses the same comment conventions as the configuration file. Key entries use a fixed format of the form

**keyno type key**

where, **keyno** is a positive integer, **type** is a single character which defines the key format, and **key** is the key itself.

The key may be given in one of three different formats, controlled by the type character. The three key types, and corresponding formats, are listed following.

**Table 13 XNTPD Key File Parameters**

Parameter	Description
S	The key is a 64-bit hexadecimal number in the format specified in the DES specification; that is, the high order seven bits of each octet are used to form the 56-bit key while the low order bit of each octet is given a value such that odd parity is maintained for the octet. Leading zeroes must be specified (i.e., the key must be exactly 16 hex digits long) and odd parity must be maintained. Hence a zero key, in standard format, would be given as 0101010101010101.
N	The key is a 64-bit hexadecimal number in the format specified in the NTP standard. This is the same as the DES format, except the bits in each octet have been rotated one bit right so that the parity bit is now the high order bit of the octet. Leading zeroes must be specified and odd parity must be maintained. A zero key in NTP format would be specified as 8080808080808080.
A	The key is a 1-to-8 character ASCII string. A key is formed from this by using the low order 7 bits of each ASCII character in the string, with zeroes added on the right when necessary to form a full width 56-bit key, in the same way that encryption keys are formed from Unix passwords.
M	The key is a 1-to-8 character ASCII string, using the MD5 authentication scheme. Note that both the keys and the authentication schemes (DES or MD5) must be identical between a set of peers sharing the same key number.



Note that the keys used by the NTPQ and XNTPDC programs are checked against passwords requested by the programs and entered by hand, so it is generally appropriate to specify these keys in ASCII format.

## Monitoring Options

XNTPD includes a comprehensive monitoring facility suitable for continuous, long-term recording of server and client timekeeping performance. See the statistics commands below for a listing and example of each type of statistics currently supported. Statistic files are managed using file generation sets and scripts in the `/scripts` directory of this distribution. Using these facilities and Unix cron jobs, the data can be automatically summarized and archived for retrospective analysis.

**Table 14** XNTPD Monitoring Command Parameters

Parameter	Description
<code>statistics <i>name</i> [ ... ]</code>	Enables writing of statistics records. Currently, three kinds of name statistics are supported.
<code>loopstats</code>	<p>Enables recording of loop filter statistics information. Each update of the local clock outputs a line of the following form to the file generation set named <code>loopstats</code>:</p> <pre>48773 10847.650 0.0001307 17.3478 2</pre> <p>The first two fields show the date (Modified Julian Day) and time (seconds and fraction past UTC midnight). The next three fields show time offset in seconds, frequency offset in parts-per-million and time constant of the clock-discipline algorithm at each update of the clock.</p>
<code>peerstats</code>	<p>Enables recording of peer statistics information. This includes statistics records of all peers of a NTP server and of special signals, where present and configured. Each valid update appends a line of the following form to the current element of a file generation set named <code>peerstats</code>:</p> <pre>48773 10847.650 127.127.4.1 9714 -0.001605 0.00000 0.00142</pre> <p>The first two fields show the date (Modified Julian Day) and time (seconds and fraction past UTC midnight). The next two fields show the peer address in dotted-quad notation and status, respectively. The status field is encoded in hex in the format. The final three fields show the offset, delay and dispersion, all in seconds.</p>
<code>clockstats</code>	<p>Enables recording of clock driver statistics information. Each update received from a clock driver outputs a line of the following form to the file generation set named <code>clockstats</code>:</p> <pre>49213 525.624 127.127.4.1 93 226 00:08:29.606 D</pre> <p>The first two fields show the date (Modified Julian Day) and time (seconds and fraction past UTC midnight). The next field shows the clock address in dotted-quad notation. The final field shows the last timecode received from the clock in decoded ASCII format, where meaningful. In some clock drivers a good deal of additional information can be gathered and displayed as well. See information specific to each clock for further details.</p>
<code>statsdir <i>directory_path</i></code>	Indicates the full path of a directory where statistics files should be created. This keyword allows the (otherwise constant) filegen filename prefix to be modified for file generation sets, which is useful for handling statistics logs.

Parameter	Description
filegen <i>name</i> [ file <i>filename</i> ] [ type <i>typename</i> ] [ flag <i>flagval</i> ] [ link   nolink ] [ enable   disable ]	Configures setting of generation file set <i>name</i> . Generation file sets provide a means for handling files that are continuously growing during the lifetime of a server. Server statistics are a typical example for such files. Generation file sets provide access to a set of files used to store the actual data. At any time at most one element of the set is being written to. The type given specifies when and how data will be directed to a new element of the set. This way, information stored in elements of a file set that are currently unused are available for administrative operations without the risk of disturbing the operation of XNTPD. (Most important: they can be removed to free space for new data produced.)

Filenames of set members are built from three elements:

**prefix:** This is a constant filename path. It is not subject to modifications via the filegen option. It is defined by the server, usually specified as a compile-time constant. It may, however, be configurable for individual file generation sets via other commands. For example, the prefix used with **loopstats** and **peerstats** generation can be configured using the **statsdir** option explained above.

**filename:** This string is directly concatenated to the prefix mentioned above (no intervening / (slash)). This can be modified using the **file** argument to the **filegen** statement. No .. elements are allowed in this component to prevent filenames referring to parts outside the filesystem hierarchy denoted by **prefix**.

**suffix:** This part reflects individual elements of a file set. It is generated according to the type of a file set.

A file generation set is characterized by its type. The following types are supported:

- ♦ **none:** The file set is actually a single plain file.
- ♦ **pid:** One element of file set is used per incarnation of a XNTPD server. This type does not perform any changes to file set members during runtime, however it provides an easy way of separating files belonging to different XNTPD server incarnations. The set member filename is built by appending a . (dot) to concatenated prefix and filename strings, and appending the decimal representation of the process ID of the XNTPD server process.
- ♦ **day:** One file generation set element is created per day. A day is defined as the period between 00:00 and 24:00 UTC. The file set member suffix consists of a . (dot) and a day specification in the form YYYYMMDD. YYYY is a 4-digit year number (e.g., 1992). MM is a two digit month number. DD is a two digit day number. Thus, all information written at 10 December 1992 would end up in a file named prefix filename.19921210.
- ♦ **week:** Any file set member contains data related to a certain week of a year. The term week is defined by computing day-of-year modulo 7. Elements of such a file generation set are distinguished by appending the following suffix to the file set filename base: A dot, a 4-digit year number, the letter W, and a 2-digit week number. For example, information from January, 10th 1992 would end up in a file with suffix .1992W1.
- ♦ **month:** One generation file set element is generated per month. The file name suffix consists of a dot, a 4-digit year number, and a 2-digit month.

Parameter	Description
filegen <i>name</i> [ file <i>filename</i> ] [ type <i>typename</i> ] [ flag <i>flagval</i> ] [ link   nolink ] [ enable   disable ] ( <i>cont'd.</i> )	<ul style="list-style-type: none"> <li>♦ <b>year:</b> One generation file element is generated per year. The filename suffix consists of a dot and a 4 digit year number.</li> <li>♦ <b>age:</b> This type of file generation sets changes to a new element of the file set every 24 hours of server operation. The filename suffix consists of a dot, the letter a, and an 8-digit number. This number is taken to be the number of seconds the server is running at the start of the corresponding 24-hour period. Information is only written to a file generation by specifying enable; output is prevented by specifying disable.</li> </ul>

It is convenient to be able to access the current element of a file generation set by a fixed name. This feature is enabled by specifying link and disabled using nolink. If link is specified, a hard link from the current file set element to a file without suffix is created. When there is already a file with this name and the number of links of this file is one, it is renamed appending a dot, the letter C, and the pid of the XNTPD server process. When the number of links is greater than one, the file is unlinked. This allows the current file to be accessed by a constant name.

## Access Control Options

XNTPD implements a general purpose address-and-mask based restriction list. The list is sorted by address and by mask, and the list is searched in this order for matches, with the last match found defining the restriction flags associated with the incoming packets. The source address of incoming packets is used for the match, with the 32-bit address being added with the mask associated with the restriction entry and then compared with the entry's address (which has also been added with the mask) to look for a match.

The restriction facility was implemented in conformance with the access policies for the original NSFnet backbone time servers. While this facility may be otherwise useful for keeping unwanted or broken remote time servers from affecting your own, it should not be considered an alternative to the standard NTP authentication facility. Source address based restrictions are easily circumvented by a determined cracker.

**Table 15** XNTPD Access Control Parameters

Parameter	Description
	<p><b>ntpport:</b> This is actually a match algorithm modifier, rather than a restriction flag. Its presence causes the restriction entry to be matched only if the source port in the packet is the standard NTP UDP port (123). Both ntpport and non-ntpport may be specified. The ntpport is considered more specific and is sorted later in the list.</p> <p>Default restriction list entries, with the flags ignore, ntpport, for each of the local host's interface addresses are inserted into the table at startup to prevent the server from attempting to synchronize to its own time. A default entry is also always present, though if it is otherwise unconfigured; no flags are associated with the default entry (i.e., everything besides your own NTP server is unrestricted).</p>
clientlimit <i>limit</i>	Set the <i>client_limit</i> variable, which limits the number of simultaneous access-controlled clients. The default value for this variable is 3.
clientperiod <i>period</i>	Set the <i>client_limit_period</i> variable, which specifies the number of seconds after which a client is considered inactive and thus no longer is counted for client limit restriction. The default value for this variable is 3600 seconds.

## Miscellaneous Options

Table 16 Miscellaneous XNTPD Parameters

Parameter	Description
<code>broadcastdelay seconds</code>	The broadcast and multicast modes require a special calibration to determine the network delay between the local and remote servers. Ordinarily, this is done automatically by the initial protocol exchanges between the local and remote servers. In some cases, the calibration procedure may fail due to network or server access controls, for example. This command specifies the default delay to be used under these circumstances. Typically (for Ethernet), a number between 0.003 and 0.007 seconds is appropriate. The default when this command is not used is 0.004 seconds.
<code>trap host_address [ port port_number ] [ interface interface_address ]</code>	<p>This command configures a trap receiver at the given host address and port number for sending messages with the specified local interface address. If the port number is unspecified, a value of 18447 is used. If the interface address is not specified, the message is sent with a source address of the local interface the message is sent through.</p> <p><b>NOTE:</b> On a multihomed host the interface used may vary from time to time with routing changes.</p> <p>The trap receiver will generally log event messages and other information from the server in a log file. While such monitor programs may also request their own trap dynamically, configuring a trap receiver will ensure that no messages are lost when the server is started.</p>
<code>setvar variable [ default ]</code>	<p>This command adds an additional system variable. These variables can be used to distribute additional information such as the access policy. If the variable of the form <i>name = value</i> is followed by the default keyword, the variable will be listed as part of the default system variables (NTPQ rv command). These additional variables serve informational purposes only. They are not related to the protocol other than they can be listed. The known protocol variables will always override any variables defined via the setvar mechanism.</p> <p>There are three special variables that contain the names of all variable of the same group. The <i>sys_var_list</i> holds the names of all system variables. The <i>peer_var_list</i> holds the names of all peer variables and the <i>clock_var_list</i> holds the names of the reference clock variables.</p>
<code>logfile logfile</code>	This command specifies the location of an alternate log file to be used instead of the default system syslog facility.

### Variables

Most variables used by the NTP protocol can be examined with the XNTPDC (mode 7 messages) and the NTPQ (mode 6 messages). Currently, very few variables can be modified via mode 6 messages. These variables are either created with the setvar directive or the leap warning bits. The leap warning bits can be set in the leapwarning variable up to one month ahead. Both the leapwarning and leapindication variables have a slightly different encoding than the usual leap bits interpretation:

- ♦ **00:** The daemon passes the leap bits of its synchronization source (usual mode of operation).
- ♦ **01/10:** A leap second is added/deleted (operator forced leap second).
- ♦ **11:** Leap information from the synchronizations source is ignored (thus LEAP\_NOWARNING is passed on).

# 6

## Monitoring and Security

This chapter describes NTP monitoring and security.

### Monitoring Time Synchronization

The quality of time synchronization can be monitored. It is based on the accuracy of the time provided by the time provider to the time consumer.

The time quality variables like offset, jitter, and precision can be measured and logged for online or offline analysis in the text mode for any NTPv3-compliant operating system.

The NTPQ utility can be used to monitor time quality variables consists of the following commands:

**Table 17** NTPQ commands

Command	Description
peers	Obtains a list of in-spec peers of the server, along with a summary of each peer's state. Summary information includes the remote peer's address, reference ID (0.0.0.0 if the refID is unknown), stratum, type (local, unicast, multicast, or broadcast); when the last packet was received; the polling interval (in seconds); the leachability register (in octal); and the current estimated delay, offset, and dispersion of the peer (all in seconds).
associations	Obtains and prints a list of association identifiers and peer statuses for in-spec peers of the server being queried. This command can also be used to check if the server is synchronized or not (If the Status column contains sys.peer, it means that server is synchronized).
rv <i>associate</i>	Obtains various time parameters from the server and displays them. The most important parameter is filter offset, which gives the last eight time offsets of the host server with the reference server.
rvi <i>index</i>	Obtains various time parameters from the server and displays them. The sequencing index can be used instead of <i>assocID</i> (as in the rv command). This makes it easy to monitor and remember the association ID every time.

For more information about the commands, see [“NTPQ” on page 34](#).

The following figure displays the output of the NTPQ peers, association, and rv *assocID* commands.

Figure 11 Output of NTPQ Peers, Association and RV AssocID Commands

```

Novell RConsole J: SHERLOCK
Server Screens NTPQ (active)
ntp> peers
=====
remote          refid      st t when poll reach  delay  offset  disp
=====
*osg-dt-7.blr.no osg-nw5-1.blr.n 3 u  93  16 377   0.26 -39.730 23.30
ntp> as
=====
ind assID status  conf reach auth condition  last_event cnt
=====
  1 36772 f664   yes yes  ok sys.peer  reachable  6
ntp> rv 36772
status=f664 reach, conf, auth, sel_sys.peer, 6 events, event_reach
srcadr=osg-dt-7.blr.novell.com, srcport=123, dstadr=164.99.159.212,
dstport=123, keyid=4, stratum=3, precision=-18, rootdelay=543.21,
rootdispersion=180.91, refid=osg-nw5-1.blr.novell.com,
reftime=c2329163.3d2a6000 Mon, Mar 31 2003 15:35:23.238,
delay= 0.26, offset= -39.73, dispersion=23.30, reach=377, valid=8,
hmode=3, pmode=4, hpoll=4, ppoll=4, leap=00, flash=0x0<OK>,
org=c232917c.f664e000 Mon, Mar 31 2003 15:35:48.999,
rec=c232917d.0a157000 Mon, Mar 31 2003 15:35:49.039,
xmt=c232917d.09fe7000 Mon, Mar 31 2003 15:35:49.039,
filtdelay= 0.26 0.23 0.26 0.26 0.26 0.24 0.27 0.38,
filtoffset= -39.73 -14.02 -16.02 -18.20 -21.27 -24.98 -29.25 -34.85,
filtererror= 0.02 0.50 0.99 1.48 1.97 2.46 2.93 3.42
ntp> _

```

The following figure displays the output of the NTPQ rvi *index* command.

Figure 12 Output of the NTPQ RVI Index Command

```

Novell RConsole J: SHERLOCK
Server Screens NTPQ (active)
ntp>
ntp>
ntp>
ntp>
ntp>
ntp>
ntp>
ntp>
ntp>
ntp>
ntp> rvi 1
status=f664 reach, conf, auth, sel_sys.peer, 6 events, event_reach
srcadr=osg-dt-7.blr.novell.com, srcport=123, dstadr=164.99.159.212,
dstport=123, keyid=4, stratum=3, precision=-18, rootdelay=542.57,
rootdispersion=181.46, refid=osg-nw5-1.blr.novell.com,
reftime=c23291c2.67fd2000 Mon, Mar 31 2003 15:36:58.406,
delay= 0.26, offset= -27.37, dispersion=9.54, reach=377, valid=8,
hmode=3, pmode=4, hpoll=4, ppoll=4, leap=00, flash=0x0<OK>,
org=c23291dd.42e7a000 Mon, Mar 31 2003 15:37:25.261,
rec=c23291dd.49f1e000 Mon, Mar 31 2003 15:37:25.288,
xmt=c23291dd.49dc4000 Mon, Mar 31 2003 15:37:25.288,
filtdelay= 0.26 0.26 0.24 0.26 0.23 0.26 0.26 0.26,
filtoffset= -27.37 -33.74 -41.08 -39.73 -14.02 -16.02 -18.20 -21.27,
filtererror= 0.02 0.50 0.99 1.48 1.97 2.46 2.94 3.43
ntp> _

```

## Health Monitor

You can monitor NTP remotely using NORM. You can monitor all the servers that are authenticated to the Novell eDirectory™ 8.7.3 tree. Select the server you want to monitor. The Peer, Associations, rv and rvi details are displayed. These details are similar to output the NTPQ peers, NTPQ associations and NTPQ read variables commands (rv and rvi) gives. For more information, see [Table 17, “NTPQ commands,” on page 53](#).

Figure 13 Monitoring NTP Using NORM

The screenshot displays the 'Monitored Server Group' window in the Health Monitor application. The left sidebar contains a navigation tree with categories like 'Diagnose Server', 'Manage Server', 'Manage Applications', 'Manage Hardware', 'Manage eDirectory', and 'Use Group Operations'. The main pane shows details for 'Host 192.168.1.1'.

**Peers**

remote	refid	st	t	when	poll	reach	delay	offset	disp
cheereo.com	78.79.86.76	5	u	-	16	3	2.18	-3109680.957	7881.33
33ntp.gns.com	0.0.0.0	16	u	-	16	0	0.00	0.000	16000.00
image.com	78.79.86.76	5	u	17	16	1	0.61	-628399	15875.02

**Associations**

id	ssid	status	conf	reach	auth	condition	last_event	cnt
1	44868	9014	yes	yes	none	insane	reachable	1
2	44869	9000	yes	no				
3	44870	9014	yes	yes	none	insane	reachable	1

**Variables**

**Association ID 44868**

```

ssid=44868
status=9014 reach, conf, 1 event, event_reach srcadr=cheereo.novell.com
srcport=123 dstadr=164.99.159.212 dstport=123
keyid=0 stratum=5 precision=-17 rootdelay=0.00
rootdispersion=0.00 refid=78.79.86.76
reftime=c27eb636.5c8e2691 Wed, May 28 2003 9:44:46.361
delay=2.18 offset=-3109680.96 dispersion=7881.33 reach=003 valid=1
hmode=3 pmode=4 hpoll=4 ppoll=6 leap=00 flash=0x0
orig=c27eb636.5c8e2691 Wed, May 28 2003 9:44:46.361
rec=c27ec25c.0b292000 Wed, May 28 2003 10:36:36.043
xmt=c27ec25c.0a999000 Wed, May 28 2003 10:36:36.041
filtdelay=2.18 1.60 0.00 0.00 0.00 0.00 0.00 0.00
filttoffset=-3109680.97 -3109668.33 0.00 0.00 0.00 0.00 0.00 0.00
filtererror=0.02 0.52 16000.00 16000.00 16000.00 16000.00 16000.00 16000.00
  
```

## Security

XNTPD supports the optional authentication procedure specified in NTP versions 2 and 3.

When an association runs in the authenticated mode, each packet transmitted appends a 32-bit key ID and a 64/128-bit cryptographic checksum of the packet contents. This is computed using either the Data Encryption Standard (DES) or Message Digest (MD5) algorithms. These algorithms provide sufficient protection from message modification attacks.

**NOTE:** Distribution of DES algorithm implementation is restricted to U.S. and Canada, while MD5 is currently free from such restrictions.

In both the algorithms, the receiving peer recomputes the checksum and compares it with the one included in the packet. For this to work, the peers must share at least one encryption key and, furthermore, must associate the shared key with the same key ID.

This requires some minor modifications to the basic packet processing procedures, as required by the specification. These modifications are enabled by the “enable authenticate” configuration declaration.

In particular, the following servers are marked untrustworthy and unsuitable for synchronization in the authenticated mode:

- ♦ Peers that send unauthenticated packets
- ♦ Peers that send authenticated packets, which the local server is unable to decrypt
- ♦ Peers that send authenticated packets encrypted using a key NTP does not trust

**NOTE:** Though the server might know many keys (identified by many key IDs), it is possible to declare only a subset of these as trusted. This allows the server to share keys with a client that trusts the server and requires authenticated time, even though the server does not trust the client.

Also, some additional configuration language is required to specify the key ID to be used to authenticate each configured peer association. For example, for a server running in authenticated mode, the configuration file might look similar to the following:

```
# peer configuration for 128.100.100.7
# (expected to operate at stratum 2)
# fully authenticated this time
peer 128.100.49.105 key 22 # suzuki.ccie.utoronto.ca
peer 128.8.10.1 key 4      # umdl.umd.edu
peer 192.35.82.50 key 6   # lilben.tn.cornell.edu
enable auth
keys sys:\etc\ntp.keys    # path for key file
trustedkey 22 4 6         # define trusted keys
requestkey 15              # key (6) for accessing server variables
controlkey 15              # key (7) for accessing server variables
authdelay 0.000094        # authentication delay (Sun4c/50 IPX)
```

- ♦ The `enable auth` line enables authentication processing.
- ♦ The `keys sys:\etc\ntp.keys` line specifies the path to the keys file (see below and the XNTPD document page for details of the file format).
- ♦ The `trustedkey` line identifies those keys that are known to be uncompromising; the remainder presumably represent the expired or possibly compromised keys. Both sets of keys must be declared by the key identifier in the `ntp.keys` file described below. This provides a way to retire old keys while minimizing the frequency of delicate key-distribution procedures.
- ♦ The `requestkey 15` line establishes the key to be used for mode-6 control messages as specified in RFC-1305 and used by the NTPQ utility program.
- ♦ The `controlkey 15` establishes the key to be used for mode-7 private control messages used by the XNTPDC utility program. These keys are used to prevent unauthorized modification of daemon variables.

As a general rule, keys should be chosen randomly, except possibly the request and control keys, which must be typed by the user as a password.

The `ntp.keys` file contains the list of keys and associated key IDs that the server knows about.

**WARNING:** This file is better left unreadable by anyone except admin.

The contents of the `ntp.keys` file might look similar to the following:



```
# ntp keys file (ntp.keys)

1    N    29233E0461ECD6AE    # des key in NTP format
2    M    RIrop8KPPvQvYotM    # md5 key as an ASCII random string
14   M    sundial              # md5 key as an ASCII string
15   A    sundial              # des key as an ASCII string

# the following 3 keys are identical

10   A    SeCReT
10   N    d3e54352e5548080
10   S    a7cb86a4cba80101
```

Each line in the key file has three attributes. For example:

```
1    N    29233E0461ECD6AE
```

In the above example:

- ♦ 1 is the key ID
- ♦ N is the key format
- ♦ 29233E0461ECD6AE is the key itself

The following table explains the four key formats.

**Table 18**     **XNTPD Security Key Formats**

Key Format	Description
A	Indicates a DES key written as a 1-to-8 character string in 7-bit ASCII representation, with each character standing for a key octet (like a UNIX password).
M	Indicates an MD5 key written as a 1-to-31 character ASCII string in the A format.
N	Indicates a DES key again written as a hex number, but in the NTP standard format with the high order bit of each octet being the (odd) parity bit.
S	Indicates a DES key written as a hex number in the DES standard format, with the low order bit (LSB) of each octet being the (odd) parity bit.

**NOTE:** Due to the simple token routine, the characters #, \t, \n, \0 and a space cannot be used in either a DES or MD5 ASCII key. Key 0 (zero) is used for special purposes and should not appear in this file.

## Sample Scenario

This sample scenario demonstrates a setup where XNTPD on ServerB (time consumer) needs to take time from the XNTPD on ServerA (time provider) with authentication.

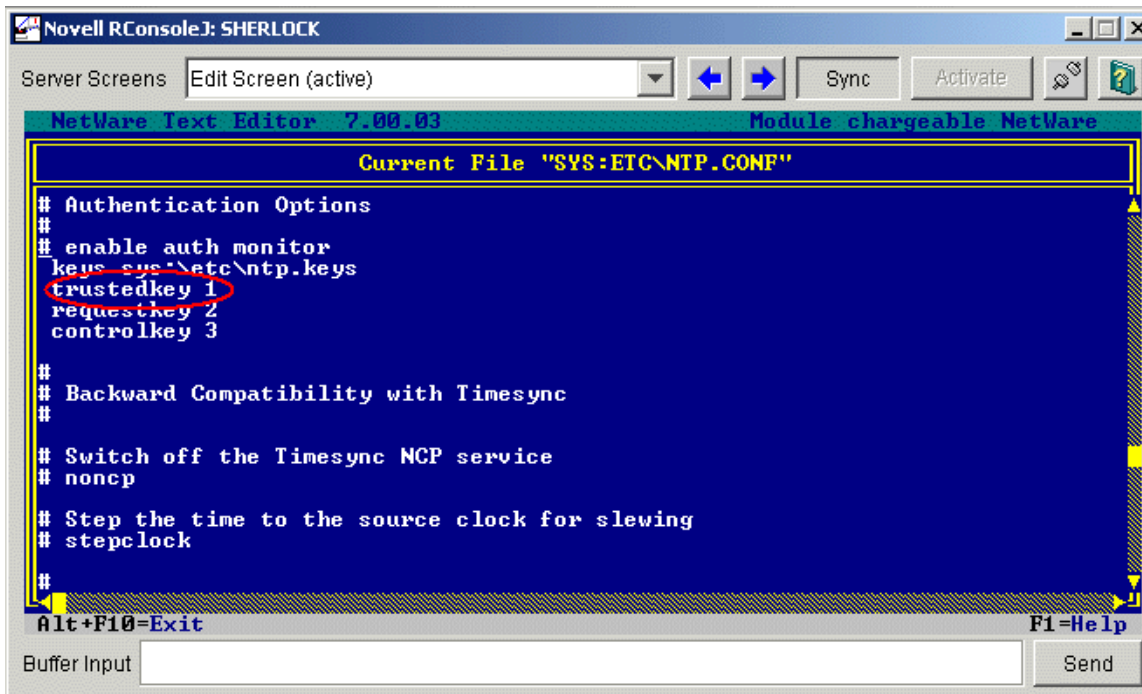
### On ServerA:

- ♦ In the ntp.conf file located in sys:\etc, make the following changes:
  - ♦ Mention sys:\etc as the path of the key file as follows:

```
keys sys:\etc\ntp.keys
```

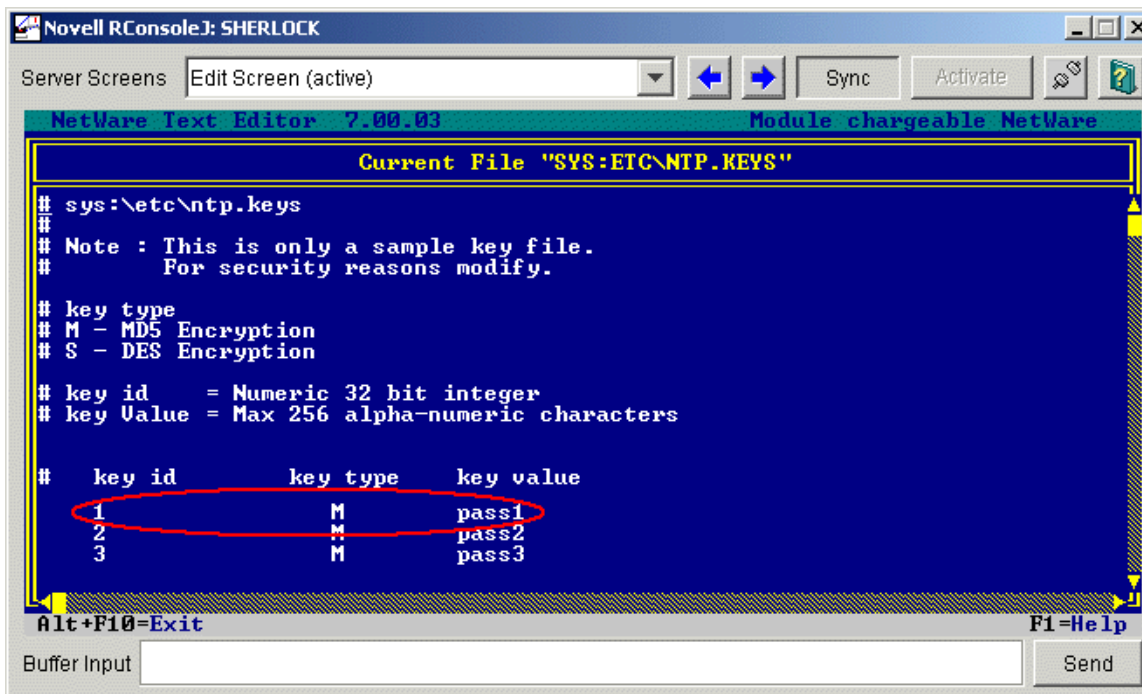
- ♦ Mark 1 as the trusted key ID. See the following figure for more details.

Figure 14 Ntp.conf File of ServerA



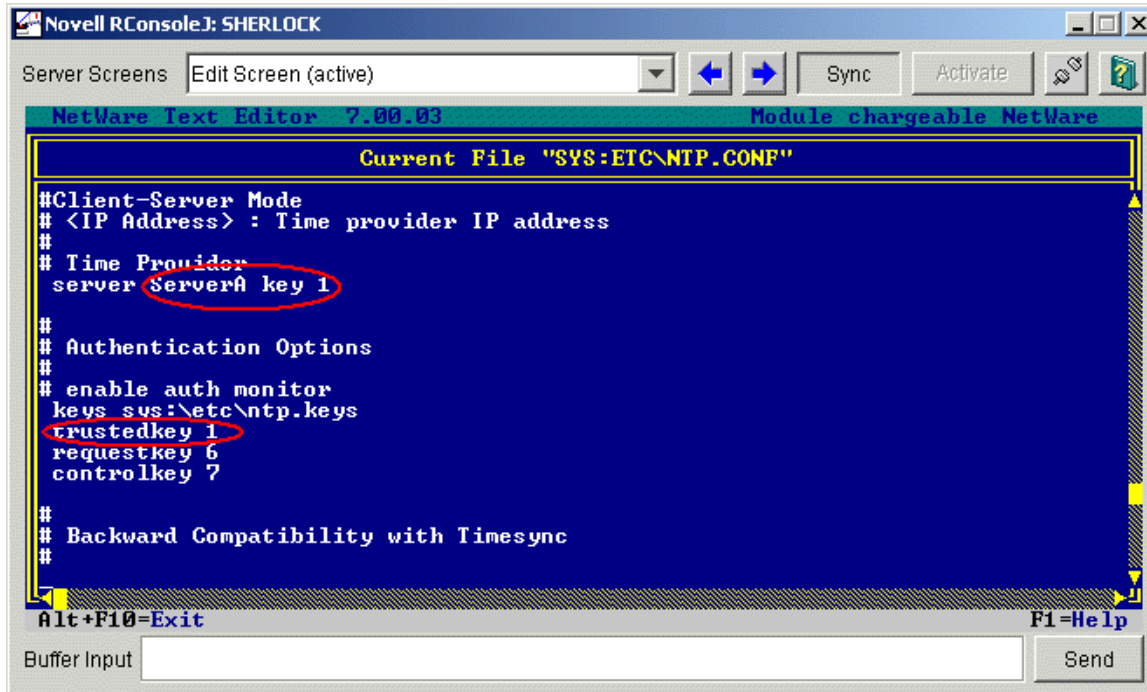
- ♦ In the ntp.keys file located in sys:\etc, give a key value for key ID 1, pass1.

Figure 15 Ntp.keys File of ServerA



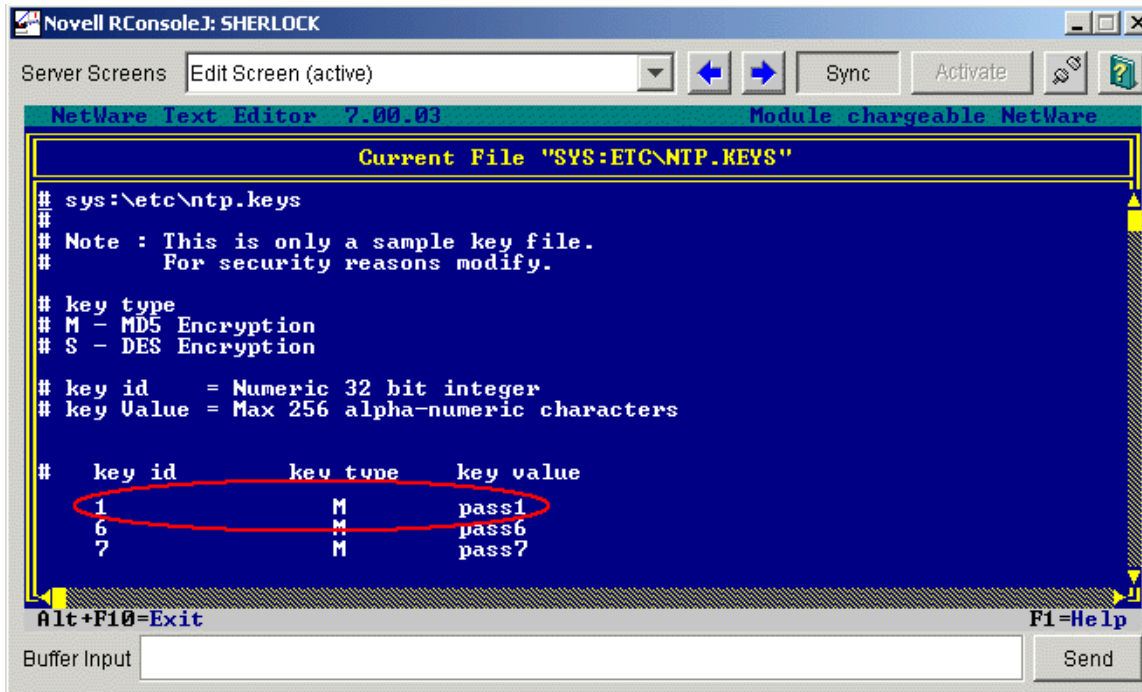
**On ServerB:**

- ♦ In the ntp.conf file located in sys:\etc, make the following changes:
  - ♦ Mention sys:\etc as the path of the key file as follows:  
`keys sys:\etc\ntp.keys`
  - ♦ Mark 1 as the trusted key ID. See the following figure for more details.

**Figure 16** Ntp.conf File of ServerB

- ♦ In the ntp.keys file located in sys:\etc, give the same key value (that was given in ServerA for key ID 1) as shown in the following figure.

Figure 17 Ntp.keys File of ServerB



```
Novell RConsoleJ: SHERLOCK
Server Screens Edit Screen (active)
NetWare Text Editor 7.00.03 Module chargeable NetWare
Current File "SYS:ETC\NTP.KEYS"
# sys:\etc\ntp.keys
#
# Note : This is only a sample key file.
#       For security reasons modify.
#
# key type
# M - MD5 Encryption
# S - DES Encryption
#
# key id    = Numeric 32 bit integer
# key Value = Max 256 alpha-numeric characters
#
# key id    key type    key value
# 1         M          pass1
# 6         M          pass6
# 7         M          pass7
Alt+F10=Exit F1=Help
Buffer Input Send
```

After entering the details in ServerA's and ServerB's ntp.conf and ntp.keys files, restart XNTPD on both the servers.

# 7

## Troubleshooting NTP

The following sections give suggestions and resources for solving issues with NTP:

- ♦ “XNTPD” on page 61
- ♦ “NTPDate” on page 63
- ♦ “General” on page 63

### XNTPD

This section provides solutions to problems you might encounter when using XNTPD.

#### XNTPD -T slp does not update ntp.conf

**Problem:** You are looking for a server that is advertising its Timesync SINGLE time source service that is in a different tree.

**Action:** To verify this:

- 1** Enter `display slp services timesync.novell` at the server console to display a list of servers advertising the Timesync SINGLE service.
- 2** If you know the tree name of your server, then enter `display slp services timesync.novell://treename==mytreename`.

#### XNTPD broadcast functionality is not working

**Problem:** You are using an incorrect subnet broadcast ID.

**Action:** Ensure that you have specified the correct subnet broadcast address.

**Action:** Use `showipconf` in NTPQ.

#### Unable to configure XNTPD remotely using XNTPDC

**Problem:** You do not have the proper keys to authenticate to the server.

**Action:** Ensure that you have the request key ID and the key of the server you are trying to configure. You can obtain the request key ID from the `ntp.conf` file and the request key from the `ntp.keys` file. Both files are located in the `sys:\etc` directory.

#### XNTPD cannot obtain time in the broadcast/multicast mode

**Problem:** You have not authenticated to broadcast/multicast server.

**Action:** Start XNTPD with the `-A` option. This disables authentication.

**Action:** Obtain the time provider’s key, copy it to `ntp.keys` file, and mark it as a trusted key in `ntp.conf`. Ensure that this key is present in the broadcast/multicast server.

## XNTPD takes a long time to synchronize

- Problem:** The polling interval is large. With a large polling interval each poll will be more far apart. XNTPD needs 5 successful polls (offset lesser then 150 ms) to transit into synchronized mode. By default, XNTPD has minpoll value set to 4. This means that every poll will be 2 power 4 seconds away. Hence, XNTPD will sync in, at best,  $5 * (2 \text{ power } 4)$  seconds.
- Action:** Minimize the polling interval using minpoll. Set minpoll equal to 4 in the ntp.conf file while giving the time provider details. For example: **server IP address minpoll 4**
- Action:** Use XNTPD with the -S option or stepclock in the ntp.conf file to synchronize within 10 seconds. For more information, see [“-S” on page 41](#) or [“stepclock” on page 47](#).

## Unable to enable the debug message with XNTPD

- Problem:** Unable to get the debug message with XNTPD.
- Action:** To enable the debug message with XNTPD, enter the following at command prompt:  
**Load XNTPD -D 4**

## Time does not synchronize if XNTPD is loaded without the -A parameter in the multicast mode

- Problem:** In the multicast mode, time does not get synchronized if XNTPD is loaded without the **-A** parameter.
- Action:** If you load XNTPD with the -A parameter, it means that you are loading XNTPD with the authentication disabled.
- By default, in the broadcast/multicast mode, XNTPD starts with authentication enabled. This is because, XNTPD can obtain time only from a server it trusts and the trust be achieved only through authentication.
- If there are no masquerading servers in the subnet/network then you can start XNTPD with authentication disabled (-A option).
- Action:** Obtain the time provider's key, copy it to the ntp.keys file, and mark it as a trusted key in the ntp.conf file. Ensure that this key is present in the broadcast/multicast server.

## Time does not synchronize in the broadcast/multicast mode it takes time from the local clock

- Problem:** A local clock is not a very reliable time source and it cannot be broadcasted or multicasted.
- Action:** Use the **prefer** keyword when using a local clock as follows:  
**server 127.127.1.0 prefer**

## XNTPD hangs while loading

- Action:** Ensure that XNTPD, NTPQ, NTPDate, and XNTPDC have unloaded successfully.

## XNTPD goes out of synch frequently

- Action:** Run XNTPD with the -S option

## XNTPD exits after 5 to 10 minutes

- Problem:** The time offset between the client and server is too large; therefore, XNTPD does not adjust the clock and it exits.
- Action:** Start XNTPD with the -S option.

## NTPDate

This section provides solutions to problems you might encounter when using NTPDate.

### **NTPDate loads normally, but does not set the clock**

Action: All the error messages appear on the Logger screen, check the error details.

### **Unable to load NTPDate when Timesync / XNTPD is running**

Problem: NTPDate was unable to bind to port 123.

Action: Load NTPDate with the -u option.

## General

This section provides solutions to generic problems, you might encounter.

### **Unable to configure NTPv3 as time client**

Action: Check if the time provider is broadcasting/multicasting its service.

Action: Check if time sources are in synchronization.

Action: In case you have localhost as loopback, ensure that you specify a higher stratum for the loopback so that it has lower precedence than the other time source.

### **Unable to open the ntp.log file using Edit**

Action: You might not be able to open the ntp.log file if it was created for the first time. Subsequent loads of XNTPD would not cause this issue.





# 8

## Frequently Asked Questions

This chapter lists frequently asked questions and their answers for NTPv3.

### How Do I Configure NTP for Fault Tolerance?

Configure two servers as follows.

#### For Server1:

- ♦ Configure XNTPD to obtain time from the external NTP time source and mark it as prefer.
- ♦ Add a local clock as the second server with a stratum value greater than external NTP source.

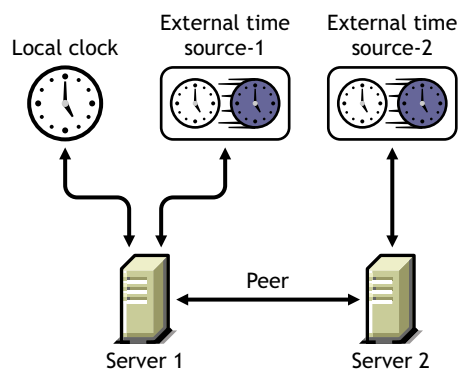
```
server IP_address_of_external_time_source-1 prefer
server 127.127.1.0
fudge 127.127.1.0 stratum 4
```

#### For Server2:

- ♦ Configure XNTPD to obtain time from the external NTP time source and mark it as prefer.
- ♦ Add Server1 as peer to Server2.

```
server IP_address_of_external_time_source-2 prefer
peer IP_address_of_Server1
```

**Figure 18 Fault Tolerance Configuration**



Server1 and Server2 will provide time to all other servers in the network.

## How Do I Find the Subnet Broadcast Address?

To find the subnet broadcast address, do one of the following:

- ♦ Use **showipconf** in NTPQ.
- ♦ Start XNTPD with debug level 4 and log option enabled. XNTPD displays the subnet broadcast address for each time source.

## How Do I Find which Server Is Selected for Synchronization?

To view the server selected for synchronization, do one of the following:

- ♦ Run NTPQ with the server name.
- ♦ Enter the command **pe**.

The server selected for synchronization is shown with an asterisk (\*).

## How Do I Find the Offset Value between the Client and the Server?

Do one of the following:

- ♦ Enter **ntpdate -d server\_name** at command prompt. This lists out the offset.
- ♦ Run NTPQ with the client and enter **pe**. This lists the offset of the client with each time source.

## Can a Single Timesync Server Take Time from a NetWare 6.5 Server Running NTPv3?

Yes, a Single Timesync server can take time from a NetWare 6.5 server running NTPv3 in both NTP and NCP modes.

## Revision History

### May 9, 2005

- ♦ Changed references of iManager 2.0 to iManager 2.5.
- ♦ Changed references of eDirectory™ to eDirectory™ 8.7.3.
- ♦ Added an appendix with Documentation Updates information.

