NAME

ntpd - NTP daemon program

SYNOPSIS

ntpd [**-flags**] [**-flag** [*value*]] [**--option-name**[[=|]*value*]] [<server1> ... <serverN>]

DESCRIPTION

The **ntpd** utility is an operating system daemon which sets and maintains the system time of day in synchronism with Internet standard time servers. It is a complete implementation of the Network Time Protocol (NTP) version 4, as defined by RFC-5905, but also retains compatibility with version 3, as defined by RFC-1305, and versions 1 and 2, as defined by RFC-1059 and RFC-1119, respectively.

The **ntpd** utility does most computations in 64-bit floating point arithmetic and does relatively clumsy 64-bit fixed point operations only when necessary to preserve the ultimate precision, about 232 picoseconds. While the ultimate precision is not achievable with ordinary workstations and networks of today, it may be required with future gigahertz CPU clocks and gigabit LANs.

Ordinarily, **ntpd** reads the ntp.conf(5) configuration file at startup time 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 may be particularly useful when the local host is to be configured as a broadcast/multicast client, with all peers being determined by listening to broadcasts at run time.

If NetInfo support is built into **ntpd**, then **ntpd** will attempt to read its configuration from the NetInfo if the default ntp.conf(5) file cannot be read and no file is specified by the **-c** option.

Various internal **ntpd** variables can be displayed and configuration options altered while the **ntpd** is running using the ntpq(8) and ntpdc(8) utility programs.

When **ntpd** starts it looks at the value of umask(2), and if zero **ntpd** will set the umask(2) to 022.

OPTIONS

-4, **--ipv4**

Force IPv4 DNS name resolution. This option must not appear in combination with any of the following options: ipv6.

Force DNS resolution of following host names on the command line to the IPv4 namespace.

-6, **--ipv6**

Force IPv6 DNS name resolution. This option must not appear in combination with any of the

following options: ipv4.

Force DNS resolution of following host names on the command line to the IPv6 namespace.

-a, **--authreq**
Require crypto authentication. This option must not appear in combination with any of the following options: authnoreq.

Require cryptographic authentication for broadcast client, multicast client and symmetric passive associations. This is the default.

-A, **--authnoreq**
Do not require crypto authentication. This option must not appear in combination with any of the following options: authreq.

Do not require cryptographic authentication for broadcast client, multicast client and symmetric passive associations. This is almost never a good idea.

-b, **--bcastsync**
Allow us to sync to broadcast servers.

-c *string*, **--configfile**=*string*
configuration file name.

The name and path of the configuration file, */etc/ntp.conf* by default.

-d, **--debug-level**
Increase debug verbosity level. This option may appear an unlimited number of times.

-D *number*, **--set-debug-level**=*number*
Set the debug verbosity level. This option may appear an unlimited number of times. This option takes an integer number as its argument.

-f *string*, **--driftfile**=*string*
frequency drift file name.

The name and path of the frequency file, */etc/ntp.drift* by default. This is the same operation as the

driftfile *driftfile* configuration specification in the */etc/ntp.conf* file.

-g, **--panicgate**
Allow the first adjustment to be Big. This option may appear an unlimited number of times.

Normally, **ntpd** exits with a message to the system log if the offset exceeds the panic threshold, which is 1000 s by default. This option allows the time to be set to any value without restriction; however, this can happen only once. If the threshold is exceeded after that, **ntpd** will exit with a message to the system log. This option can be used with the **-q** and **-x** options. See the **tinker** configuration file directive for other options.

-G, **--force-step-once**
Step any initial offset correction..

Normally, **ntpd** steps the time if the time offset exceeds the step threshold, which is 128 ms by default, and otherwise slews the time. This option forces the initial offset correction to be stepped, so the highest time accuracy can be achieved quickly. However, this may also cause the time to be stepped back so this option must not be used if applications requiring monotonic time are running. See the **tinker** configuration file directive for other options.

-i *string*, **--jaildir**=*string*
Jail directory.

Chroot the server to the directory *jaildir* This option also implies that the server attempts to drop root privileges at startup. You may need to also specify a **-u** option. This option is only available if the OS supports adjusting the clock without full root privileges. This option is supported under NetBSD (configure with **--enable-clockctl**) or Linux (configure with **--enable-linuxcaps**) or Solaris (configure with **--enable-solarisprivs**).

-I *iface*, **--interface**=*iface*
Listen on an interface name or address. This option may appear an unlimited number of times.

Open the network address given, or all the addresses associated with the given interface name. This option may appear multiple times. This option also implies not opening other addresses, except wildcard and localhost. This option is deprecated. Please consider using the configuration file **interface** command, which is more versatile.

-k *string*, **--keyfile**=*string*
path to symmetric keys.

Specify the name and path of the symmetric key file. */etc/ntp.keys* is the default. This is the same operation as the **keys** *keyfile* configuration file directive.

-l *string*, **--logfile**=*string*
path to the log file.

Specify the name and path of the log file. The default is the system log file. This is the same operation as the **logfile** *logfile* configuration file directive.

-L, **--novirtualips**
Do not listen to virtual interfaces.

Do not listen to virtual interfaces, defined as those with names containing a colon. This option is deprecated. Please consider using the configuration file **interface** command, which is more versatile.

-M, **--modifymmtimer**
Modify Multimedia Timer (Windows only).

Set the Windows Multimedia Timer to highest resolution. This ensures the resolution does not change while ntpd is running, avoiding timekeeping glitches associated with changes.

-n, **--nofork**
Do not fork. This option must not appear in combination with any of the following options: wait-sync.

-N, **--nice**
Run at high priority.

To the extent permitted by the operating system, run **ntpd** at the highest priority.

-p *string*, **--pidfile**=*string*
path to the PID file.

Specify the name and path of the file used to record **ntpd**'s process ID. This is the same operation as the **pidfile** *pidfile* configuration file directive.

-P *number*, **--priority**=*number*
Process priority. This option takes an integer number as its argument.

To the extent permitted by the operating system, run **ntpd** at the specified **sched_setscheduler(SCHED_FIFO)** priority.

-q, **--quit**
Set the time and quit. This option must not appear in combination with any of the following options: saveconfigquit, wait-sync.

ntpd will not daemonize and will exit after the clock is first synchronized. This behavior mimics that of the **ntpdate** program, which will soon be replaced with a shell script. The **-g** and **-x** options can be used with this option. Note: The kernel time discipline is disabled with this option.

-r *string*, **--propagationdelay**=*string*
Broadcast/propagation delay.

Specify the default propagation delay from the broadcast/multicast server to this client. This is necessary only if the delay cannot be computed automatically by the protocol.

--saveconfigquit=*string*
Save parsed configuration and quit. This option must not appear in combination with any of the following options: quit, wait-sync.

Cause **ntpd** to parse its startup configuration file and save an equivalent to the given filename and exit. This option was designed for automated testing.

-s *string*, **--statsdir**=*string*
Statistics file location.

Specify the directory path for files created by the statistics facility. This is the same operation as the **statsdir** *statsdir* configuration file directive.

-t *tkey*, **--trustedkey**=*tkey*
Trusted key number. This option may appear an unlimited number of times.

Add the specified key number to the trusted key list.

-u *string*, **--user**=*string*
Run as userid (or userid:groupid).

Specify a user, and optionally a group, to switch to. This option is only available if the OS supports adjusting the clock without full root privileges. This option is supported under NetBSD (configure

with **--enable-clockctl**) or Linux (configure with **--enable-linuxcaps**) or Solaris (configure with **--enable-solarisprivs**).

-U *number*, **--updateinterval**=*number*
interval in seconds between scans for new or dropped interfaces. This option takes an integer number as its argument.

Give the time in seconds between two scans for new or dropped interfaces. For systems with routing socket support the scans will be performed shortly after the interface change has been detected by the system. Use 0 to disable scanning. 60 seconds is the minimum time between scans.

--var=*nvar*
make ARG an ntp variable (RW). This option may appear an unlimited number of times.

--dvar=*ndvar*
make ARG an ntp variable (RW|DEF). This option may appear an unlimited number of times.

-w *number*, **--wait-sync**=*number*
Seconds to wait for first clock sync. This option must not appear in combination with any of the following options: nofork, quit, saveconfigquit. This option takes an integer number as its argument.

If greater than zero, alters **ntpd**'s behavior when forking to daemonize. Instead of exiting with status 0 immediately after the fork, the parent waits up to the specified number of seconds for the child to first synchronize the clock. The exit status is zero (success) if the clock was synchronized, otherwise it is **ETIMEDOUT**. This provides the option for a script starting **ntpd** to easily wait for the first set of the clock before proceeding.

-x, **--slew**
Slew up to 600 seconds.

Normally, the time is slewed if the offset is less than the step threshold, which is 128 ms by default, and stepped if above the threshold. This option sets the threshold to 600 s, which is well within the accuracy window to set the clock manually. Note: Since the slew rate of typical Unix kernels is limited to 0.5 ms/s, each second of adjustment requires an amortization interval of 2000 s. Thus, an adjustment as much as 600 s will take almost 14 days to complete. This option can be used with the **-g** and **-q** options. See the **tinker** configuration file directive for other options. Note: The kernel time discipline is disabled with this option.

--usepcc
Use CPU cycle counter (Windows only).

Attempt to substitute the CPU counter for **QueryPerformanceCounter**. The CPU counter and **QueryPerformanceCounter** are compared, and if they have the same frequency, the CPU counter (RDTSC on x86) is used directly, saving the overhead of a system call.

--pccfreq=*string*
Force CPU cycle counter use (Windows only).

Force substitution the CPU counter for **QueryPerformanceCounter**. The CPU counter (RDTSC on x86) is used unconditionally with the given frequency (in Hz).

-m, **--mdns**
Register with mDNS as a NTP server.

Registers as an NTP server with the local mDNS server which allows the server to be discovered via mDNS client lookup.

-?, **--help**
Display usage information and exit.

-!, **--more-help**
Pass the extended usage information through a pager.

--version [{*v*|*c*|*n*}]
Output version of program and exit. The default mode is 'v', a simple version. The 'c' mode will print copyright information and 'n' will print the full copyright notice.

OPTION PRESETS

Any option that is not marked as *not presettable* may be preset by loading values from environment variables named:
NTPD_<option-name> or **NTPD**

USAGE

How NTP Operates

The **ntpd** utility operates by exchanging messages with one or more configured servers over a range of designated poll intervals. When started, whether for the first or subsequent times, the program requires several exchanges from the majority of these servers so the signal processing and mitigation algorithms can accumulate and groom the data and set the clock. In order to protect the network from bursts, the

initial poll interval for each server is delayed an interval randomized over a few seconds. At the default initial poll interval of 64s, several minutes can elapse before the clock is set. This initial delay to set the clock can be safely and dramatically reduced using the **iburst** keyword with the **server** configuration command, as described in ntp.conf(5).

Most operating systems and hardware of today incorporate a time-of-year (TOY) chip to maintain the time during periods when the power is off. When the machine is booted, the chip is used to initialize the operating system time. After the machine has synchronized to a NTP server, the operating system corrects the chip from time to time. In the default case, if **ntpd** detects that the time on the host is more than 1000s from the server time, **ntpd** assumes something must be terribly wrong and the only reliable action is for the operator to intervene and set the clock by hand. (Reasons for this include there is no TOY chip, or its battery is dead, or that the TOY chip is just of poor quality.) This causes **ntpd** to exit with a panic message to the system log. The **-g** option overrides this check and the clock will be set to the server time regardless of the chip time (up to 68 years in the past or future -- this is a limitation of the NTPv4 protocol). However, and to protect against broken hardware, such as when the CMOS battery fails or the clock counter becomes defective, once the clock has been set an error greater than 1000s will cause **ntpd** to exit anyway.

Under ordinary conditions, **ntpd** adjusts the clock in small steps so that the timescale is effectively continuous and without discontinuities. Under conditions of extreme network congestion, the roundtrip delay jitter can exceed three seconds and the synchronization distance, which is equal to one-half the roundtrip delay plus error budget terms, can become very large. The **ntpd** algorithms discard sample offsets exceeding 128 ms, unless the interval during which no sample offset is less than 128 ms exceeds 900s. The first sample after that, no matter what the offset, steps the clock to the indicated time. In practice this reduces the false alarm rate where the clock is stepped in error to a vanishingly low incidence.

As the result of this behavior, once the clock has been set it very rarely strays more than 128 ms even under extreme cases of network path congestion and jitter. Sometimes, in particular when **ntpd** is first started without a valid drift file on a system with a large intrinsic drift the error might grow to exceed 128 ms, which would cause the clock to be set backwards if the local clock time is more than 128 s in the future relative to the server. In some applications, this behavior may be unacceptable. There are several solutions, however. If the **-x** option is included on the command line, the clock will never be stepped and only slew corrections will be used. But this choice comes with a cost that should be carefully explored before deciding to use the **-x** option. The maximum slew rate possible is limited to 500 parts-per-million (PPM) as a consequence of the correctness principles on which the NTP protocol and algorithm design are based. As a result, the local clock can take a long time to converge to an acceptable offset, about 2,000 s for each second the clock is outside the acceptable range. During this interval the local clock will not be consistent with any other network clock and the system cannot be used for distributed applications that require correctly synchronized network time.

In spite of the above precautions, sometimes when large frequency errors are present the resulting time offsets stray outside the 128-ms range and an eventual step or slew time correction is required. If following such a correction the frequency error is so large that the first sample is outside the acceptable range, **ntpd** enters the same state as when the *ntp.drift* file is not present. The intent of this behavior is to quickly correct the frequency and restore operation to the normal tracking mode. In the most extreme cases (the host **time.ien.it** comes to mind), there may be occasional step/slew corrections and subsequent frequency corrections. It helps in these cases to use the **burst** keyword when configuring the server, but ONLY when you have permission to do so from the owner of the target host.

Finally, in the past many startup scripts would run ntpdate(8) or sntp(8) to get the system clock close to correct before starting ntpd(8), but this was never more than a mediocre hack and is no longer needed. If you are following the instructions in *Starting NTP (Best Current Practice)* and you still need to set the system time before starting **ntpd**, please open a bug report and document what is going on, and then look at using sntp(8) if you really need to set the clock before starting **ntpd**.

There is a way to start ntpd(8) that often addresses all of the problems mentioned above.

Starting NTP (Best Current Practice)

First, use the **iburst** option on your **server** entries.

If you can also keep a good *ntp.drift* file then ntpd(8) will effectively "warm-start" and your system's clock will be stable in under 11 seconds' time.

As soon as possible in the startup sequence, start ntpd(8) with at least the **-g** and perhaps the **-N** options. Then, start the rest of your "normal" processes. This will give ntpd(8) as much time as possible to get the system's clock synchronized and stable.

Finally, if you have processes like **dovecot** or database servers that require monotonically-increasing time, run ntp-wait(1ntp-waitmdoc) as late as possible in the boot sequence (perhaps with the **-v** flag) and after ntp-wait(1ntp-waitmdoc) exits successfully it is as safe as it will ever be to start any process that require stable time.

Frequency Discipline

The **ntpd** behavior at startup depends on whether the frequency file, usually *ntp.drift*, exists. This file contains the latest estimate of clock frequency error. When the **ntpd** is started and the file does not exist, the **ntpd** enters a special mode designed to quickly adapt to the particular system clock oscillator time and frequency error. This takes approximately 15 minutes, after which the time and frequency are set to nominal values and the **ntpd** enters normal mode, where the time and frequency are continuously tracked relative to the server. After one hour the frequency file is created and the current frequency offset written to it. When the **ntpd** is started and the file does exist, the **ntpd** frequency is initialized from the

file and enters normal mode immediately. After that the current frequency offset is written to the file at hourly intervals.

Operating Modes

The **ntpd** utility can operate in any of several modes, including symmetric active/passive, client/server broadcast/multicast and manycast, as described in the "Association Management" page (available as part of the HTML documentation provided in */usr/share/doc/ntp*). It normally operates continuously while monitoring for small changes in frequency and trimming the clock for the ultimate precision. However, it can operate in a one-time mode where the time is set from an external server and frequency is set from a previously recorded frequency file. A broadcast/multicast or manycast client can discover remote servers, compute server-client 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.

By default, **ntpd** runs in continuous mode where each of possibly several external servers is polled at intervals determined by an intricate state machine. The state machine measures the incidental roundtrip delay jitter and oscillator frequency wander and determines the best poll interval using a heuristic algorithm. Ordinarily, and in most operating environments, the state machine will start with 64s intervals and eventually increase in steps to 1024s. A small amount of random variation is introduced in order to avoid bunching at the servers. In addition, should a server become unreachable for some time, the poll interval is increased in steps to 1024s in order to reduce network overhead.

In some cases it may not be practical for **ntpd** to run continuously. A common workaround has been to run the ntpdate(8) or sntp(8) programs from a cron(8) job at designated times. However, these programs do not have the crafted signal processing, error checking or mitigation algorithms of **ntpd**. The **-q** option is intended for this purpose. Setting this option will cause **ntpd** to exit just after setting the clock for the first time. The procedure for initially setting the clock is the same as in continuous mode; most applications will probably want to specify the **iburst** keyword with the **server** configuration command. With this keyword a volley of messages are exchanged to groom the data and the clock is set in about 10 s. If nothing is heard after a couple of minutes, the daemon times out and exits. After a suitable period of mourning, the ntpdate(8) program will be retired.

When kernel support is available to discipline the clock frequency, which is the case for stock Solaris, Tru64, Linux and FreeBSD, a useful feature is available to discipline the clock frequency. First, **ntpd** is run in continuous mode with selected servers in order to measure and record the intrinsic clock frequency offset in the frequency file. It may take some hours for the frequency and offset to settle down. Then the **ntpd** is stopped and run in one-time mode as required. At each startup, the frequency is read from the file and initializes the kernel frequency.

Poll Interval Control

This version of NTP includes an intricate state machine to reduce the network load while maintaining a quality of synchronization consistent with the observed jitter and wander. There are a number of ways to tailor the operation in order enhance accuracy by reducing the interval or to reduce network overhead by increasing it. However, the user is advised to carefully consider the consequences of changing the poll adjustment range from the default minimum of 64 s to the default maximum of 1,024 s. The default minimum can be changed with the **tinker minpoll** command to a value not less than 16 s. This value is used for all configured associations, unless overridden by the **minpoll** option on the configuration command. Note that most device drivers will not operate properly if the poll interval is less than 64 s and that the broadcast server and manycast client associations will also use the default, unless overridden.

In some cases involving dial up or toll services, it may be useful to increase the minimum interval to a few tens of minutes and maximum interval to a day or so. Under normal operation conditions, once the clock discipline loop has stabilized the interval will be increased in steps from the minimum to the maximum. However, this assumes the intrinsic clock frequency error is small enough for the discipline loop correct it. The capture range of the loop is 500 PPM at an interval of 64s decreasing by a factor of two for each doubling of interval. At a minimum of 1,024 s, for example, the capture range is only 31 PPM. If the intrinsic error is greater than this, the drift file *ntp.drift* will have to be specially tailored to reduce the residual error below this limit. Once this is done, the drift file is automatically updated once per hour and is available to initialize the frequency on subsequent daemon restarts.

The huff-n'-puff Filter

In scenarios where a considerable amount of data are to be downloaded or uploaded over telephone modems, timekeeping quality can be seriously degraded. This occurs because the differential delays on the two directions of transmission can be quite large. In many cases the apparent time errors are so large as to exceed the step threshold and a step correction can occur during and after the data transfer is in progress.

The huff-n'-puff filter is designed to correct the apparent time offset in these cases. It depends on knowledge of the propagation delay when no other traffic is present. In common scenarios this occurs during other than work hours. The filter maintains a shift register that remembers the minimum delay over the most recent interval measured usually in hours. Under conditions of severe delay, the filter corrects the apparent offset using the sign of the offset and the difference between the apparent delay and minimum delay. The name of the filter reflects the negative (huff) and positive (puff) correction, which depends on the sign of the offset.

The filter is activated by the **tinker** command and **huffpuff** keyword, as described in ntp.conf(5).

ENVIRONMENT

See **OPTION PRESETS** for configuration environment variables.

FILES

/etc/ntp.conf
the default name of the configuration file
/etc/ntp.drift the default name of the drift file
/etc/ntp.keys
the default name of the key file

EXIT STATUS

One of the following exit values will be returned:

0 (EXIT_SUCCESS)
Successful program execution.

1 (EXIT_FAILURE)
The operation failed or the command syntax was not valid.

70 (EX_SOFTWARE)
libopts had an internal operational error. Please report it to autogen-users@lists.sourceforge.net. Thank you.

SEE ALSO

ntp.conf(5), ntpdate(8), ntpdc(8), ntpq(8), sntp(8)

In addition to the manual pages provided, comprehensive documentation is available on the world wide web at http://www.ntp.org/. A snapshot of this documentation is available in HTML format in */usr/share/doc/ntp*.

David L. Mills, *Network Time Protocol (Version 1)*, RFC1059.

David L. Mills, *Network Time Protocol (Version 2)*, RFC1119.

David L. Mills, *Network Time Protocol (Version 3)*, RFC1305.

David L. Mills, J. Martin, Ed., J. Burbank, and W. Kasch, *Network Time Protocol Version 4: Protocol and Algorithms Specification*, RFC5905.

David L. Mills and B. Haberman, Ed., *Network Time Protocol Version 4: Autokey Specification*, RFC5906.

H. Gerstung, C. Elliott, and B. Haberman, Ed., *Definitions of Managed Objects for Network Time*

Protocol Version 4: (NTPv4), RFC5907.

R. Gayraud and B. Lourdelet, *Network Time Protocol (NTP) Server Option for DHCPv6*, RFC5908.

AUTHORS

The University of Delaware and Network Time Foundation

COPYRIGHT

BUGS

The **ntpd** utility has gotten rather fat. While not huge, it has gotten larger than might be desirable for an elevated-priority **ntpd** running on a workstation, particularly since many of the fancy features which consume the space were designed more with a busy primary server, rather than a high stratum workstation in mind.

Please send bug reports to: http://bugs.ntp.org, bugs@ntp.org

NOTES

Portions of this document came from FreeBSD.

This manual page was *AutoGen*-erated from the **ntpd** option definitions.