NAME

ntpdc — vendor-specific NTPD control program

SYNOPSIS

ntpdc [**-flags**] [**-flag** [*value*]] [**--option-name**[[=|]*value*]] [host ...]

DESCRIPTION

ntpdc is deprecated. Please use ntpq(1) instead – it can do everything **ntpdc** used to do, and it does so using a much more sane interface.

ntpdc is a utility program used to query ntpd(8) about its current state and to request changes in that state. It uses NTP mode 7 control message formats described in the source code. The program may be run either in interactive mode or controlled using command line arguments. Extensive state and statistics information is available through the **ntpdc** interface. In addition, nearly all the configuration options which can be specified at startup using ntpd's configuration file may also be specified at run time using **ntpdc**.

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.

-c *cmd*, **--command**=*cmd*
run a command and exit. This option may appear an unlimited number of times.

The following argument is interpreted as an interactive format command and is added to the list of commands to be executed on the specified host(s).

-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.

-i, **--interactive**
Force ntpq to operate in interactive mode. This option must not appear in combination with any of the following options: command, listpeers, peers, showpeers.

Force ntpq to operate in interactive mode. Prompts will be written to the standard output and commands read from the standard input.

-l, **--listpeers**
Print a list of the peers. This option must not appear in combination with any of the following options: command.

Print a list of the peers known to the server as well as a summary of their state. This is equivalent to the 'listpeers' interactive command.

-n, **--numeric**
numeric host addresses.

Output all host addresses in dotted–quad numeric format rather than converting to the canonical host names.

-p, **--peers**
Print a list of the peers. This option must not appear in combination with any of the following options: command.

Print a list of the peers known to the server as well as a summary of their state. This is equivalent to the 'peers' interactive command.

-s, **--showpeers**
Show a list of the peers. This option must not appear in combination with any of the following options: command.

Print a list of the peers known to the server as well as a summary of their state. This is equivalent to the 'dmpeers' interactive command.

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

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

-> [*cfgfile*], **--save-opts** [=*cfgfile*]
Save the option state to *cfgfile*. The default is the *last* configuration file listed in the **OPTION PRESETS** section, below. The command will exit after updating the config file.

-< *cfgfile*, **--load-opts**=*cfgfile*, **--no-load-opts**
Load options from *cfgfile*. The *no−load−opts* form will disable the loading of earlier config/rc/ini files. *−−no−load−opts* is handled early, out of order.

--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 configuration ("RC" or ".INI") file(s) and values from environment variables named:
NTPDC_<option−name> or **NTPDC**
The environmental presets take precedence (are processed later than) the configuration files. The *homerc* files are "*$HOME*", and ".". If any of these are directories, then the file *.ntprc* is searched for within those directories.

USAGE

If one or more request options are included on the command line when **ntpdc** is executed, each of the requests will be 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, **ntpdc** will attempt to read commands from the standard input and execute 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. The **ntpdc** utility will prompt for commands if the standard input is a terminal device.

The **ntpdc** utility uses NTP mode 7 packets to communicate with the NTP server, and hence can be used to query any compatible server on the network which permits it. Note that since NTP is a UDP protocol this communication will be somewhat unreliable, especially over large distances in terms of network topology. The **ntpdc** utility makes no attempt to retransmit requests, and will time requests out if the remote host is not heard from within a suitable timeout time.

The operation of **ntpdc** are specific to the particular implementation of the `ntpd`(8) daemon and can be expected to work only with this and maybe some previous versions of the daemon. Requests from a remote **ntpdc** utility which 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.

Note that in contexts where a host name is expected, a **−4** qualifier preceding the host name forces DNS resolution to the IPv4 namespace, while a **−6** qualifier forces DNS resolution to the IPv6 namespace. Specifying a command line option other than **−i** or **−n** will cause the specified query (queries) to be sent to the indicated host(s) immediately. Otherwise, **ntpdc** will attempt to read interactive format commands from the standard input.

Interactive 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 **ntpdc** utility itself and do not result in NTP mode 7 requests being sent to a server. These are described following.

? *command_keyword*

help *command_keyword*
: A '**?**' will print a list of all the command keywords known to this incarnation of **ntpdc**. A '**?**' followed by a command keyword will print function and usage information about the command. This command is probably a better source of information about `ntpq`(1) than this manual page.

delay *milliseconds*
: 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 *hostname*
: 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 *keyid*
: 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.

quit Exit **ntpdc**.

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.

timeout *milliseconds*
Specify a timeout period for responses to server queries. The default is about 8000 milliseconds. Note that since **ntpdc** retries each query once after a timeout, the total waiting time for a timeout will be twice the timeout value set.

Control Message Commands

Query commands result in NTP mode 7 packets containing requests for information being sent to the server. These are read−only commands in that they make no modification of the server configuration state.

listpeers
Obtains and prints a brief list of the peers for which the server is maintaining state. These should include all configured peer associations as well as those peers whose stratum is such that they are considered by the server to be possible future synchronization candidates.

peers Obtains a list of peers for which the server is maintaining state, along with a summary of that state. Summary information includes the address of the remote peer, the local interface address (0.0.0.0 if a local address has yet to be determined), the stratum of the remote peer (a stratum of 16 indicates the remote peer is unsynchronized), the polling interval, in seconds, the reachability register, in octal, and the current estimated delay, offset and dispersion of the peer, all in seconds.

The character in the left margin indicates the mode this peer entry is operating in. A ‘+’ denotes symmetric active, a ‘−’ indicates symmetric passive, a ‘=’ means the remote server is being polled in client mode, a ‘^’ indicates that the server is broadcasting to this address, a ‘~’ denotes that the remote peer is sending broadcasts and a ‘~’ denotes that the remote peer is sending broadcasts and a ‘*’ marks the peer the server is currently synchronizing to.

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**(*implementation_number*, *parameter*). On **hostnames no** only IP−addresses will be displayed.

dmpeers
A slightly different peer summary list. Identical to the output of the **peers** command, except for the character in the leftmost column. Characters only appear beside peers which were included in the final stage of the clock selection algorithm. A ‘**.**’ indicates that this peer was cast off in the falseticker detection, while a ‘+’ indicates that the peer made it through. A ‘*’ denotes the peer the server is currently synchronizing with.

showpeer *peer_address* [**...**]
Shows a detailed display of the current peer variables for one or more peers. Most of these values are described in the NTP Version 2 specification.

pstats *peer_address* [**...**]
Show per−peer statistic counters associated with the specified peer(s).

clockstat *clock_peer_address* [**...**]
Obtain and print information concerning a peer clock. The values obtained provide information on the setting of fudge factors and other clock performance information.

kerninfo
Obtain and print kernel phase−lock loop operating parameters. This information is available only if the kernel has been specially modified for a precision timekeeping function.

loopinfo [**oneline** | **multiline**]
Print the values of selected loop filter variables. The loop filter is the part of NTP which deals with adjusting the local system clock. The ‘offset’ is the last offset given to the loop filter by the packet processing code. The ‘frequency’ is the frequency error of the local clock in parts−per−million (ppm). The ‘time_const’ controls the stiffness of the phase−lock loop and thus the speed at which it can adapt to oscillator drift. The ‘watchdog timer’ value is the number of seconds which have elapsed since the last sample offset was given to the loop filter. The **oneline** and **multiline** options specify the format in which this information is to be printed, with **multiline** as the default.

sysinfo
Print a variety of system state variables, i.e., state related to the local server. All except the last four lines are described in the NTP Version 3 specification, RFC−1305.

The ‘system flags’ show various system flags, some of which can be set and cleared by the **enable** and **disable** configuration commands, respectively. These are the **auth**, **bclient**, **monitor**, **pll**, **pps** and **stats** flags. See the ntpd(8) documentation for the meaning of these flags. There are two additional flags which are read only, the **kernel_pll** and **kernel_pps**. These flags indicate the synchronization status when the precision time kernel modifications are in use. The ‘kernel_pll’ indicates that the local clock is being disciplined by the kernel, while the ‘kernel_pps’ indicates the kernel discipline is provided by the PPS signal.

The ‘stability’ is the residual frequency error remaining after the system frequency correction is applied and is intended for maintenance and debugging. In most architectures, this value will initially decrease from as high as 500 ppm to a nominal value in the range .01 to 0.1 ppm. If it remains high for some time after starting the daemon, something may be wrong with the local clock, or the value of the kernel variable *kern.clockrate.tick* may be incorrect.

The ‘broadcastdelay’ shows the default broadcast delay, as set by the **broadcastdelay** configuration command.

The ‘authdelay’ shows the default authentication delay, as set by the **authdelay** configuration command.

sysstats
Print statistics counters maintained in the protocol module.

memstats
Print statistics counters related to memory allocation code.

iostats
Print statistics counters maintained in the input−output module.

timerstats
Print statistics counters maintained in the timer/event queue support code.

reslist
Obtain and print the server’s restriction list. This list is (usually) printed in sorted order and may help to understand how the restrictions are applied.

monlist [*version*]
Obtain and print traffic counts collected and maintained by the monitor facility. The version number should not normally need to be specified.

clkbug *clock_peer_address* [*...*]
Obtain debugging information for a reference clock driver. This information is provided only by some clock drivers and is mostly undecodable without a copy of the driver source in hand.

Runtime Configuration Requests

All requests which cause state changes in the server are authenticated by the server using a configured NTP key (the facility can also be disabled by the server by not configuring a key). The key number and the corresponding key must also be made known to **ntpdc**. This can be done using the **keyid** and **passwd** commands, the latter of which will prompt at the terminal for a password to use as the encryption key. You will also be prompted automatically for both the key number and password the first time a command which would result in an authenticated request to the server is given. Authentication not only provides verification that the requester has permission to make such changes, but also gives an extra degree of protection again transmission errors.

Authenticated requests always include a timestamp in the packet data, which is included in the computation of the authentication code. This timestamp is compared by the server to its receive time stamp. If they differ by more than a small amount the request is rejected. This is done for two reasons. First, it makes simple replay attacks on the server, by someone who might be able to overhear traffic on your LAN, much more difficult. Second, it makes it more difficult to request configuration changes to your server from topologically remote hosts. While the reconfiguration facility will work well with a server on the local host, and may work adequately between time−synchronized hosts on the same LAN, it will work very poorly for more distant hosts. As such, if reasonable passwords are chosen, care is taken in the distribution and protection of keys and appropriate source address restrictions are applied, the run time reconfiguration facility should provide an adequate level of security.

The following commands all make authenticated requests.

addpeer *peer_address* [*keyid*] [*version*] [**prefer**]
: Add a configured peer association at the given address and operating in symmetric active mode. Note that an existing association with the same peer may be deleted when this command is executed, or may simply be converted to conform to the new configuration, as appropriate. If the optional *keyid* is a nonzero integer, all outgoing packets to the remote server will have an authentication field attached encrypted with this key. If the value is 0 (or not given) no authentication will be done. The *version* can be 1, 2 or 3 and defaults to 3. The **prefer** keyword indicates a preferred peer (and thus will be used primarily for clock synchronisation if possible). The preferred peer also determines the validity of the PPS signal − if the preferred peer is suitable for synchronisation so is the PPS signal.

addserver *peer_address* [*keyid*] [*version*] [**prefer**]
: Identical to the addpeer command, except that the operating mode is client.

broadcast *peer_address* [*keyid*] [*version*] [**prefer**]
: Identical to the addpeer command, except that the operating mode is broadcast. In this case a valid key identifier and key are required. The *peer_address* parameter can be the broadcast address of the local network or a multicast group address assigned to NTP. If a multicast address, a multicast−capable kernel is required.

unconfig *peer_address* [**. . .**]
: This command causes the configured bit to be removed from the specified peer(s). In many cases this will cause the peer association to be deleted. When appropriate, however, the association may persist in an unconfigured mode if the remote peer is willing to continue on in this fashion.

fudge *peer_address* [**time1**] [**time2**] [*stratum*] [*refid*]
: This command provides a way to set certain data for a reference clock. See the source listing for further information.

enable [**auth** | **bclient** | **calibrate** | **kernel** | **monitor** | **ntp** | **pps** | **stats**]

disable [**auth** | **bclient** | **calibrate** | **kernel** | **monitor** | **ntp** | **pps** | **stats**]
These commands operate in the same way as the **enable** and **disable** configuration file commands of `ntpd`(8).

auth Enables the server to synchronize with unconfigured peers only if the peer has been correctly authenticated using either public key or private key cryptography. The default for this flag is enable.

bclient
Enables the server to listen for a message from a broadcast or multicast server, as in the multicastclient command with default address. The default for this flag is disable.

calibrate
Enables the calibrate feature for reference clocks. The default for this flag is disable.

kernel Enables the kernel time discipline, if available. The default for this flag is enable if support is available, otherwise disable.

monitor
Enables the monitoring facility. See the documentation here about the **monlist** command or further information. The default for this flag is enable.

ntp Enables time and frequency discipline. In effect, this switch opens and closes the feedback loop, which is useful for testing. The default for this flag is enable.

pps Enables the pulse−per−second (PPS) signal when frequency and time is disciplined by the precision time kernel modifications. See the "A Kernel Model for Precision Timekeeping" (available as part of the HTML documentation provided in `/usr/share/doc/ntp`) page for further information. The default for this flag is disable.

stats Enables the statistics facility. See the **Monitoring Options** section of `ntp.conf`(5) for further information. The default for this flag is disable.

restrict *address mask flag* [. . .]
This command operates in the same way as the **restrict** configuration file commands of `ntpd`(8).

unrestrict *address mask flag* [. . .]
Unrestrict the matching entry from the restrict list.

delrestrict *address mask* [**ntpport**]
Delete the matching entry from the restrict list.

readkeys
Causes the current set of authentication keys to be purged and a new set to be obtained by rereading the keys file (which must have been specified in the `ntpd`(8) configuration file). This allows encryption keys to be changed without restarting the server.

trustedkey *keyid* [. . .]

untrustedkey *keyid* [. . .]
These commands operate in the same way as the **trustedkey** and **untrustedkey** configuration file commands of `ntpd`(8).

authinfo
Returns information concerning the authentication module, including known keys and counts of encryptions and decryptions which have been done.

`traps` Display the traps set in the server. See the source listing for further information.

`addtrap` *`address`* [*`port`*] [*`interface`*]
Set a trap for asynchronous messages. See the source listing for further information.

`clrtrap` *`address`* [*`port`*] [*`interface`*]
Clear a trap for asynchronous messages. See the source listing for further information.

`reset` Clear the statistics counters in various modules of the server. See the source listing for further information.

ENVIRONMENT

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

FILES

See **OPTION PRESETS** for configuration files.

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.

66 (EX_NOINPUT)
A specified configuration file could not be loaded.

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), `ntpd`(8)

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

AUTHORS

The formatting directives in this document came from FreeBSD.

COPYRIGHT

Copyright (C) 1992–2017 The University of Delaware and Network Time Foundation all rights reserved. This program is released under the terms of the NTP license, <http://ntp.org/license>.

BUGS

The **`ntpdc`** utility is a crude hack. Much of the information it shows is deadly boring and could only be loved by its implementer. The program was designed so that new (and temporary) features were easy to hack in, at great expense to the program's ease of use. Despite this, the program is occasionally useful.

Please report bugs to http://bugs.ntp.org .

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

NOTES

This manual page was *AutoGen*–erated from the **ntpdc** option definitions.