NAME

ntpq — standard NTP query program

SYNOPSIS

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

DESCRIPTION

The **ntpq** utility program is used to query NTP servers to monitor NTP operations and performance, requesting information about current state and/or changes in that state. The program may be run either in interactive mode or controlled using command line arguments. Requests to read and write arbitrary variables can be assembled, with raw and pretty−printed output options being available. The **ntpq** utility can also obtain and print a list of peers in a common format by sending multiple queries to the server.

If one or more request options is included on the command line when **ntpq** 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, **ntpq** 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 **ntpq** utility will prompt for commands if the standard input is a terminal device.

ntpq uses NTP mode 6 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 **ntpq** utility makes one attempt to retransmit requests, and will time requests out if the remote host is not heard from within a suitable timeout time.

Note that in contexts where a host name is expected, a **−4** qualifier preceding the host name forces resolution to the IPv4 namespace, while a **−6** qualifier forces resolution to the IPv6 namespace. For examples and usage, see the “NTP Debugging Techniques” page.

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, **ntpq** will attempt to read interactive format commands from the standard input.

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.

A number of interactive format commands are executed entirely within the **ntpq** utility itself and do not result in NTP requests being sent to a server. These are described following.

? [*command*]

help [*command*] A ‘?’ by itself will print a list of all the commands known to **ntpq**. A ‘?’ followed by a command name will print function and usage information about the command.

addvars *name*[=*value*][,...]

rmvars *name*[,...]

clearvars

showvars The arguments to this command consist of a list of items of the form *name*[=*value*], where the =*value* is ignored, and can be omitted, in requests to the server to read variables. The **ntpq** utility maintains an internal list in which data to be included in messages can be assembled, and displayed or set 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 **clearvars** command removes all variables from the list. The **showvars** command displays the current list of optional variables.

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. The command **authenticate** causes **ntpq** to display whether or not it is currently authenticating requests.

cooked Causes output from query commands to be "cooked", so that variables which are recognized by **ntpq** will have their values reformatted for human consumption. Variables which **ntpq** could not decode completely are marked with a trailing ‘?’.

debug [**more**|**less**|**off**]
With no argument, displays the current debug level. Otherwise, the debugging level is changed as indicated.

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. Without any arguments, displays the current delay.

drefid [**hash**|**ipv4**]
Display refids as IPv4 or hash. Without any arguments, displays whether refids are shown as IPv4 addresses or hashes.

exit Exit **ntpq**.

host [*name*] Set the host to which future queries will be sent. The *name* may be either a host name or a numeric address. Without any arguments, displays the current host.

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. Without any arguments, displays whether host names or numeric addresses are shown.

keyid [*keyid*] This command allows the specification of a key number to be used to authenticate configuration requests. This must correspond to the **controlkey** key number the server has been configured to use for this purpose. Without any arguments, displays the current *keyid*.

keytype [*digest*]
Specify the digest algorithm to use for authenticating requests, with default **MD5**. If **ntpq** was built with OpenSSL support, and OpenSSL is installed, *digest* can be any message digest algorithm supported by OpenSSL. If no argument is given, the current **keytype** *digest* algorithm used is displayed.

ntpversion [**1**|**2**|**3**|**4**]
Sets the NTP version number which **ntpq** claims in packets. Defaults to 3, and 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. With no argument, displays the current NTP version that will be used when communicating with servers.

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.

poll [*n*] [**verbose**]
Poll an NTP server in client mode *n* times. Poll not implemented yet.

quit Exit **ntpq**.

raw Causes all output from query commands is printed as received from the remote server. The only formating/interpretation done on the data is to transform nonascii data into a printable (but barely understandable) form.

timeout [*milliseconds*]
Specify a timeout period for responses to server queries. The default is about 5000 milliseconds. Without any arguments, displays the current timeout period. 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.

version Display the version of the **ntpq** program.

Control Message Commands

Association ids are used to identify system, peer and clock variables. System variables are assigned an association id of zero and system name space, while each association is assigned a nonzero association id and peer namespace. Most control commands send a single message to the server and expect a single response message. The exceptions are the **peers** command, which sends a series of messages, and the **mreadlist** and **mreadvar** commands, which iterate over a range of associations.

apeers Display a list of peers in the form:

```
[tally]remote refid assid st t when pool reach delay
offset jitter
```

where the output is just like the **peers** command except that the **refid** is displayed in hex format and the association number is also displayed.

associations
Display a list of mobilized associations in the form:

```
ind assid status conf reach auth condition last_event
cnt
```

Variable	Description
ind	index on this list
assid	association id
status	peer status word
conf	**yes**: persistent, **no**: ephemeral
reach	**yes**: reachable, **no**: unreachable
auth	**ok**, **yes**, **bad** and **none**
condition	selection status (see the **select** field of the peer status word)
last_event	event report (see the **event** field of the peer status word)
cnt	event count (see the **count** field of the peer status word)

authinfo Display the authentication statistics counters: time since reset, stored keys, free keys, key lookups, keys not found, uncached keys, expired keys, encryptions, decryptions.

clocklist [*associd*]

cl [*associd*]
Display all clock variables in the variable list for those associations supporting a reference clock.

clockvar [*associd*] [*name*[=*value*]] [,...]
cv [*associd*] [*name*[=*value*]] [,...]
Display a list of clock variables for those associations supporting a reference clock.

:config *configuration command line*
Send the remainder of the command line, including whitespace, to the server as a run–time configuration command in the same format as a line in the configuration file. This command is experimental until further notice and clarification. Authentication is of course required.

config-from-file *filename*
Send each line of *filename* to the server as run–time configuration commands in the same format as lines in the configuration file. This command is experimental until further notice and clarification. Authentication is required.

ifstats Display status and statistics counters for each local network interface address: interface number, interface name and address or broadcast, drop, flag, ttl, mc, received, sent, send failed, peers, uptime. Authentication is required.

iostats Display network and reference clock I/O statistics: time since reset, receive buffers, free receive buffers, used receive buffers, low water refills, dropped packets, ignored packets, received packets, packets sent, packet send failures, input wakeups, useful input wakeups.

kerninfo Display kernel loop and PPS statistics: associd, status, pll offset, pll frequency, maximum error, estimated error, kernel status, pll time constant, precision, frequency tolerance, pps frequency, pps stability, pps jitter, calibration interval, calibration cycles, jitter exceeded, stability exceeded, calibration errors. As with other ntpq output, times are in milliseconds; very small values may be shown as exponentials. The precision value displayed is in milliseconds as well, unlike the precision system variable.

lassociations
Perform the same function as the associations command, except display mobilized and unmobilized associations, including all clients.

lopeers [**-4**|**-6**]
Display a list of all peers and clients showing **dstadr** (associated with the given IP version).

lpassociations
Display the last obtained list of associations, including all clients.

lpeers [**-4**|**-6**]
Display a list of all peers and clients (associated with the given IP version).

monstats Display monitor facility status, statistics, and limits: enabled, addresses, peak addresses, maximum addresses, reclaim above count, reclaim older than, kilobytes, maximum kilobytes.

mreadlist *associdlo associdhi*
mrl *associdlo associdhi*
Perform the same function as the **readlist** command for a range of association ids.

mreadvar *associdlo associdhi* [*name*] [,...]
This range may be determined from the list displayed by any command showing associations.

mrv *associdlo associdhi* [*name*] [,...]
Perform the same function as the **readvar** command for a range of association ids. This range may be determined from the list displayed by any command showing associations.

mrulist [**limited** | **kod** | **mincount**=*count* | **laddr**=*localaddr* | **sort**=[−]*sortorder* | **resany**=*hexmask* | **resall**=*hexmask*]

Display traffic counts of the most recently seen source addresses collected and maintained by the monitor facility. With the exception of **sort**=[−]*sortorder*, the options filter the list returned by ntpd(8). The **limited** and **kod** options return only entries representing client addresses from which the last packet received triggered either discarding or a KoD response. The **mincount**=*count* option filters entries representing less than *count* packets. The **laddr**=*localaddr* option filters entries for packets received on any local address other than *localaddr*. **resany**=*hexmask* and **resall**=*hexmask* filter entries containing none or less than all, respectively, of the bits in *hexmask*, which must begin with **0x**. The *sortorder* defaults to **lstint** and may be **addr**, **avgint**, **count**, **lstint**, or any of those preceded by '−' to reverse the sort order. The output columns are:

Column Description

lstint Interval in seconds between the receipt of the most recent packet from this address and the completion of the retrieval of the MRU list by **ntpq**.

avgint Average interval in s between packets from this address.

rstr Restriction flags associated with this address. Most are copied unchanged from the matching **restrict** command, however 0x400 (kod) and 0x20 (limited) flags are cleared unless the last packet from this address triggered a rate control response.

r Rate control indicator, either a period, **L** or **K** for no rate control response, rate limiting by discarding, or rate limiting with a KoD response, respectively.

m Packet mode.

v Packet version number.

count Packets received from this address.

rport Source port of last packet from this address.

remote address
host or DNS name, numeric address, or address followed by claimed DNS name which could not be verified in parentheses.

opeers [**−4** | **−6**]

Obtain and print the old−style list of all peers and clients showing **dstadr** (associated with the given IP version), rather than the **refid**.

passociations

Perform the same function as the **associations** command, except that it uses previously stored data rather than making a new query.

peers Display a list of peers in the form:

```
[tally]remote refid st t when pool reach delay offset
jitter
```

Variable Description

[tally] single−character code indicating current value of the **select** field of the *peer status word*: **decode.html#peer**

remote host name (or IP number) of peer. The value displayed will be truncated to 15 characters unless the **ntpq −w** option is given, in which case the full value will be displayed on the first line, and if too long, the remaining data will be displayed on the next line.

refid source IP address or *'kiss code*: **decode.html#kiss**

st stratum: 0 for local reference clocks, 1 for servers with local reference clocks, ..., 16 for unsynchronized server clocks

t **u**: unicast or manycast client, **b**: broadcast or multicast client, **p**: pool source, **l**: local (reference clock), **s**: symmetric (peer), **A**: manycast server, **B**: broadcast server, **M**: multicast server

when time in seconds, minutes, hours, or days since the last packet was received, or '−' if a packet has never been received

poll poll interval (s)

reach reach shift register (octal)

delay roundtrip delay

offset offset of server relative to this host

jitter offset RMS error estimate.

pstats *associd*

Display the statistics for the peer with the given *associd*: associd, status, remote host, local address, time last received, time until next send, reachability change, packets sent, packets received, bad authentication, bogus origin, duplicate, bad dispersion, bad reference time, candidate order.

readlist [*associd*]

rl [*associd*]

Display all system or peer variables. If the *associd* is omitted, it is assumed to be zero.

readvar [*associd name*[=*value*] [, ...]]

rv [*associd name*[=*value*] [, ...]]

Display the specified system or peer variables. If *associd* is zero, the variables are from the **System Variables** name space, otherwise they are from the **Peer Variables** name space. The *associd* is required, as the same name can occur in both spaces. If no *name* is included, all operative variables in the name space are displayed. In this case only, if the *associd* is omitted, it is assumed to be zero. Multiple names are specified with comma separators and without whitespace. Note that time values are represented in milliseconds and frequency values in parts−per−million (PPM). Some NTP timestamps are represented in the format *YYYYMM DD TTTT*, where *YYYY* is the year, *MM* the month of year, *DD* the day of month and *TTTT* the time of day.

reslist Display the access control (restrict) list for **ntpq**. Authentication is required.

saveconfig *filename*

Save the current configuration, including any runtime modifications made by **:config** or **config−from−file**, to the NTP server host file *filename*. This command will be rejected by the server unless *saveconfigdir*: **miscopt.html#saveconfigdir** appears in the ntpd(8) configuration file. *filename* can use date(1) format specifiers to substitute the current date and time, for example,

```
saveconfig ntp-%Y%m%d-%H%M%S.conf.
```

The filename used is stored in system variable **savedconfig**. Authentication is required.

sysinfo Display system operational summary: associd, status, system peer, system peer mode, leap indicator, stratum, log2 precision, root delay, root dispersion, reference id, reference time, system jitter, clock jitter, clock wander, broadcast delay, symm. auth. delay.

sysstats Display system uptime and packet counts maintained in the protocol module: uptime, sysstats reset, packets received, current version, older version, bad length or format, authentication failed, declined, restricted, rate limited, KoD responses, processed for time.

timerstats
Display interval timer counters: time since reset, timer overruns, calls to transmit.

writelist *associd*
Set all system or peer variables included in the variable list.

writevar *associd name*=*value* [, ...]
Set the specified variables in the variable list. If the *associd* is zero, the variables are from the **System Variables** name space, otherwise they are from the **Peer Variables** name space. The *associd* is required, as the same name can occur in both spaces. Authentication is required.

Status Words and Kiss Codes

The current state of the operating program is shown in a set of status words maintained by the system. Status information is also available on a per–association basis. These words are displayed by the **readlist** and **associations** commands both in hexadecimal and in decoded short tip strings. The codes, tips and short explanations are documented on the *Event Messages and Status Words*: **decode.html** page. The page also includes a list of system and peer messages, the code for the latest of which is included in the status word.

Information resulting from protocol machine state transitions is displayed using an informal set of ASCII strings called *kiss codes*: **decode.html#kiss**. The original purpose was for kiss–o'–death (KoD) packets sent by the server to advise the client of an unusual condition. They are now displayed, when appropriate, in the reference identifier field in various billboards.

System Variables

The following system variables appear in the **readlist** billboard. Not all variables are displayed in some configurations.

Variable	Description
status	*system status word*: **decode.html#sys**
version	NTP software version and build time
processor	hardware platform and version
system	operating system and version
leap	leap warning indicator (0–3)
stratum	stratum (1–15)
precision	precision (log2 s)
rootdelay	total roundtrip delay to the primary reference clock
rootdisp	total dispersion to the primary reference clock
refid	reference id or *kiss code*: **decode.html#kiss**
reftime	reference time
clock	date and time of day
peer	system peer association id
tc	time constant and poll exponent (log2 s) (3–17)
mintc	minimum time constant (log2 s) (3–10)
offset	combined offset of server relative to this host
frequency	frequency drift (PPM) relative to hardware clock
sys_jitter	combined system jitter
clk_wander	clock frequency wander (PPM)
clk_jitter	clock jitter

tai	TAI−UTC offset (s)
leapsec	NTP seconds when the next leap second is/was inserted
expire	NTP seconds when the NIST leapseconds file expires

The jitter and wander statistics are exponentially−weighted RMS averages. The system jitter is defined in the NTPv4 specification; the clock jitter statistic is computed by the clock discipline module.

When the NTPv4 daemon is compiled with the OpenSSL software library, additional system variables are displayed, including some or all of the following, depending on the particular Autokey dance:

Variable	Description
host	Autokey host name for this host
ident	Autokey group name for this host
flags	host flags (see Autokey specification)
digest	OpenSSL message digest algorithm
signature	OpenSSL digest/signature scheme
update	NTP seconds at last signature update
cert	certificate subject, issuer and certificate flags
until	NTP seconds when the certificate expires

Peer Variables

The following peer variables appear in the **readlist** billboard for each association. Not all variables are displayed in some configurations.

Variable	Description
associd	association id
status	*peer status word*: **decode.html#peer**
srcadr	source (remote) IP address
srcport	source (remote) port
dstadr	destination (local) IP address
dstport	destination (local) port
leap	leap indicator (0−3)
stratum	stratum (0−15)
precision	precision (log2 s)
rootdelay	total roundtrip delay to the primary reference clock
rootdisp	total root dispersion to the primary reference clock
refid	reference id or *kiss code*: **decode.html#kiss**
reftime	reference time
rec	last packet received time
reach	reach register (octal)
unreach	unreach counter
hmode	host mode (1−6)
pmode	peer mode (1−5)
hpoll	host poll exponent (log2 s) (3−17)
ppoll	peer poll exponent (log2 s) (3−17)
headway	headway (see *Rate Management and the Kiss−o'−Death Packet*: **rate.html**)
flash	*flash status word*: **decode.html#flash**
keyid	symmetric key id
offset	filter offset
delay	filter delay
dispersion	filter dispersion

jitter	filter jitter
bias	unicast/broadcast bias
xleave	interleave delay (see *NTP Interleaved Modes*: **xleave.html**)

The **bias** variable is calculated when the first broadcast packet is received after the calibration volley. It represents the offset of the broadcast subgraph relative to the unicast subgraph. The **xleave** variable appears only for the interleaved symmetric and interleaved modes. It represents the internal queuing, buffering and transmission delays for the preceding packet.

When the NTPv4 daemon is compiled with the OpenSSL software library, additional peer variables are displayed, including the following:

Variable	Description
flags	peer flags (see Autokey specification)
host	Autokey server name
flags	peer flags (see Autokey specification)
signature	OpenSSL digest/signature scheme
initsequence	initial key id
initkey	initial key index
timestamp	Autokey signature timestamp
ident	Autokey group name for this association

Clock Variables

The following clock variables appear in the **clocklist** billboard for each association with a reference clock. Not all variables are displayed in some configurations.

Variable	Description
associd	association id
status	*clock status word*: **decode.html#clock**
device	device description
timecode	ASCII time code string (specific to device)
poll	poll messages sent
noreply	no reply
badformat	bad format
baddata	bad date or time
fudgetime1	fudge time 1
fudgetime2	fudge time 2
stratum	driver stratum
refid	driver reference id
flags	driver flags

OPTIONS

-4, **--ipv4**

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

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

-6, **--ipv6**

Force IPv6 name resolution. This option must not appear in combination with any of the following options: ipv4.

Force 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, peers.

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

−n, **−−numeric**

numeric host addresses.

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

−−old−rv

Always output status line with readvar.

By default, **ntpq** now suppresses the **associd=...** line that precedes the output of **readvar** (alias **rv**) when a single variable is requested, such as **ntpq −c "rv 0 offset"**. This option causes **ntpq** to include both lines of output for a single−variable **readvar**. Using an environment variable to preset this option in a script will enable both older and newer **ntpq** to behave identically in this regard.

−p, **−−peers**

Print a list of the peers. This option must not appear in combination with any of the following options: interactive.

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.

−r *keyword*, **−−refid**=*keyword*

Set default display type for S2+ refids. This option takes a keyword as its argument. The argument sets an enumeration value that can be tested by comparing them against the option value macro. The available keywords are:

hash ipv4
or their numeric equivalent.

The default *keyword* for this option is:

ipv4

Set the default display format for S2+ refids.

-w, **--wide**
Display the full 'remote' value.

Display the full value of the 'remote' value. If this requires more than 15 characters, display the full value, emit a newline, and continue the data display properly indented on the next line.

-?, **--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:
NTPQ_<option−name> or **NTPQ**
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.

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.

AUTHORS

The University of Delaware and Network Time Foundation

COPYRIGHT

BUGS

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

NOTES

This manual page was *AutoGen*−erated from the **ntpq** option definitions.