

# The `dig` command

Hans Kruse and Carl Bruggeman

Jan 7, 2007

## The `dig` command

The `dig` command is only found on modern Unix systems. Windows contains the predecessor to `dig`, `nslookup`. `dig` is used to make queries into the Domain Name System (DNS). We will use `dig` to convert names to addresses without using an application like a web browser (which might not give you good information about why a lookup might have failed). We also use `dig` to initiate queries that usually happen “behind the scenes”, such as looking up names for addresses (a “reverse” query), or looking up name servers for an organization. This document only shows very simple examples; check the man page for all the other options.

## What does the `dig` output mean?

Below is a sample of what the `dig` command might produce:

```
>dig pavel.its.ohiou.edu

; <<>> DiG 9.2.2 <<>> pavel.its.ohiou.edu
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<-  opcode: QUERY, status: NOERROR, id: 6551
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 6, ADDITIONAL: 6

;; QUESTION SECTION:
pavel.its.ohiou.edu.      IN      A

;; ANSWER SECTION:
pavel.its.ohiou.edu.     86400   IN      A      132.235.67.21

;; AUTHORITY SECTION:
its.OhioU.edu.           86400   IN      NS     watson.cns.OhioU.edu.
its.OhioU.edu.           86400   IN      NS     holmes.cns.OhioU.edu.
its.OhioU.edu.           86400   IN      NS     boss.cs.OhioU.edu.
its.OhioU.edu.           86400   IN      NS     oucsace.cs.OhioU.edu.
its.OhioU.edu.           86400   IN      NS     dns2.cso.uiuc.edu.
its.OhioU.edu.           86400   IN      NS     dns1.cso.uiuc.edu.

;; ADDITIONAL SECTION:
watson.cns.OhioU.edu.    86400   IN      A      132.235.64.1
holmes.cns.OhioU.edu.    86400   IN      A      132.235.64.2
boss.cs.OhioU.edu.       28800   IN      A      132.235.1.1
```

```
oucsace.cs.OhioU.edu. 28800 IN A 132.235.1.2
dns2.cso.uiuc.edu. 171639 IN A 128.174.5.104
dns1.cso.uiuc.edu. 171639 IN A 128.174.5.103
```

```
;; Query time: 1 msec
;; SERVER: 132.235.64.1#53(132.235.64.1)
;; WHEN: Sun Jan 7 14:55:23 2007
;; MSG SIZE rcvd: 299
```

You can see that the output contains several distinct sections:

- **global options:** starts a section that contains mainly program settings used for the query, and some information about the result. Look for the **status:** field, it should read **NOERROR**. Failed queries will still produce output very similar to the one above, but with a different status, such as **NXDOMAIN** for “non-existent domain name”. The **ANSWER:** field should show a number greater than zero.
- The **QUESTION** section restates your query in DNS syntax.
- The **ANSWER** section contains one or more DNS records that correspond to your query. We will talk about these in a later lab. The answer here says that **pavel.its.ohiou.edu** corresponds to an Internet (**IN**) address (**A**), namely 132.235.67.21; this answer is valid for 86,400 seconds (i.e. 1 day).
- The remaining sections tell you which name servers have authority over the name you just looked up, and they helpfully provide the IP addresses for these servers in case you want to query them directly. Finally, the command prints some statistics about the performance of the query.

# The dig man page from Mac OS X

DIG(1) DIG(1)

## NAME

dig - DNS lookup utility

## SYNOPSIS

```
dig [ @server ] [ -b address ] [ -c class ] [ -f filename ] [ -k
filename ] [ -p port# ] [ -t type ] [ -x addr ] [ -y name:key ] [
name ] [ type ] [ class ] [ queryopt... ]
```

```
dig [ -h ]
```

```
dig [ global-queryopt... ] [ query... ]
```

## DESCRIPTION

dig (domain information groper) is a flexible tool for interrogating DNS name servers. It performs DNS lookups and displays the answers that are returned from the name server(s) that were queried. Most DNS administrators use dig to troubleshoot DNS problems because of its flexibility, ease of use and clarity of output. Other lookup tools tend to have less functionality than dig.

Although dig is normally used with command-line arguments, it also has a batch mode of operation for reading lookup requests from a file. A brief summary of its command-line arguments and options is printed when the -h option is given. Unlike earlier versions, the BIND9 implementation of dig allows multiple lookups to be issued from the command line.

Unless it is told to query a specific name server, dig will try each of the servers listed in /etc/resolv.conf.

When no command line arguments or options are given, will perform an NS query for "." (the root).

## SIMPLE USAGE

A typical invocation of dig looks like:

```
dig @server name type
```

where:

server is the name or IP address of the name server to query. This can be an IPv4 address in dotted-decimal notation or an IPv6 address in colon-delimited notation. When the supplied server argument is a hostname, dig resolves that name before querying that name server. If no server argument is provided, dig consults /etc/resolv.conf and queries the name servers listed there. The

reply from the name server that responds is displayed.

`name` is the name of the resource record that is to be looked up.

`type` indicates what type of query is required -- ANY, A, MX, SIG, etc. `type` can be any valid query type. If no `type` argument is supplied, dig will perform a lookup for an A record.

## OPTIONS

The `-b` option sets the source IP address of the query to `address`. This must be a valid address on one of the host's network interfaces.

The default query class (IN for internet) is overridden by the `-c` option. `class` is any valid class, such as HS for Hesiod records or CH for CHAOSNET records.

The `-f` option makes dig operate in batch mode by reading a list of lookup requests to process from the file `filename`. The file contains a number of queries, one per line. Each entry in the file should be organised in the same way they would be presented as queries to dig using the command-line interface.

If a non-standard port number is to be queried, the `-p` option is used. `port#` is the port number that dig will send its queries instead of the standard DNS port number 53. This option would be used to test a name server that has been configured to listen for queries on a non-standard port number.

The `-t` option sets the query type to `type`. It can be any valid query type which is supported in BIND9. The default query type "A", unless the `-x` option is supplied to indicate a reverse lookup. A zone transfer can be requested by specifying a type of AXFR. When an incremental zone transfer (IXFR) is required, `type` is set to `ixfr=N`. The incremental zone transfer will contain the changes made to the zone since the serial number in the zone's SOA record was N.

Reverse lookups - mapping addresses to names - are simplified by the `-x` option. `addr` is an IPv4 address in dotted-decimal notation, or a colon-delimited IPv6 address. When this option is used, there is no need to provide the name, class and type arguments. dig automatically performs a lookup for a name like `11.12.13.10.in-addr.arpa` and sets the query type and class to PTR and IN respectively. By default, IPv6 addresses are looked up using the IP6.ARPA domain and binary labels as defined in RFC2874. To use the older RFC1886 method using the IP6.INT domain and "nibble" labels, specify the `-n` (nibble) option.

To sign the DNS queries sent by dig and their responses using transaction signatures (TSIG), specify a TSIG key file using the `-k` option. You can also specify the TSIG key itself on the command line using the `-y` option; `name` is the name of the TSIG key and `key` is the actual key. The key is a base-64 encoded string, typically generated by `dnssec-key-`

gen(8). Caution should be taken when using the -y option on multi-user systems as the key can be visible in the output from ps(1) or in the shell's history file. When using TSIG authentication with dig, the name server that is queried needs to know the key and algorithm that is being used. In BIND, this is done by providing appropriate key and server statements in named.conf.

## QUERY OPTIONS

dig provides a number of query options which affect the way in which lookups are made and the results displayed. Some of these set or reset flag bits in the query header, some determine which sections of the answer get printed, and others determine the timeout and retry strategies.

Each query option is identified by a keyword preceded by a plus sign (+). Some keywords set or reset an option. These may be preceded by the string no to negate the meaning of that keyword. Other keywords assign values to options like the timeout interval. They have the form +keyword=value. The query options are:

`+[no]tcp`

Use [do not use] TCP when querying name servers. The default behaviour is to use UDP unless an AXFR or IXFR query is requested, in which case a TCP connection is used.

`+[no]vc`

Use [do not use] TCP when querying name servers. This alternate syntax to +[no]tcp is provided for backwards compatibility. The "vc" stands for "virtual circuit".

`+[no]ignore`

Ignore truncation in UDP responses instead of retrying with TCP. By default, TCP retries are performed.

`+domain=somename`

Set the search list to contain the single domain somename, as if specified in a domain directive in /etc/resolv.conf, and enable search list processing as if the +search option were given.

`+[no]search`

Use [do not use] the search list defined by the searchlist or domain directive in resolv.conf (if any). The search list is not used by default.

`+[no]defname`

Deprecated, treated as a synonym for +[no]search

`+[no]aaonly`

This option does nothing. It is provided for compatibility with old versions of dig where it set an unimplemented resolver flag.

`+[no]adflag`

Set [do not set] the AD (authentic data) bit in the query. The AD bit currently has a standard meaning only in responses, not in queries, but the ability to set the bit in the query is provided for completeness.

`+[no]cdflag`

Set [do not set] the CD (checking disabled) bit in the query. This requests the server to not perform DNSSEC validation of responses.

`+[no]recursive`

Toggle the setting of the RD (recursion desired) bit in the query. This bit is set by default, which means dig normally sends recursive queries. Recursion is automatically disabled when the `+nssearch` or `+trace` query options are used.

`+[no]nssearch`

When this option is set, dig attempts to find the authoritative name servers for the zone containing the name being looked up and display the SOA record that each name server has for the zone.

`+[no]trace`

Toggle tracing of the delegation path from the root name servers for the name being looked up. Tracing is disabled by default. When tracing is enabled, dig makes iterative queries to resolve the name being looked up. It will follow referrals from the root servers, showing the answer from each server that was used to resolve the lookup.

`+[no]cmd`

toggles the printing of the initial comment in the output identifying the version of dig and the query options that have been applied. This comment is printed by default.

`+[no]short`

Provide a terse answer. The default is to print the answer in a verbose form.

`+[no]identify`

Show [or do not show] the IP address and port number that supplied the answer when the `+short` option is enabled. If short form answers are requested, the default is not to show the source address and port number of the server that provided the answer.

`+[no]comments`

Toggle the display of comment lines in the output. The default is to print comments.

`+[no]stats`

This query option toggles the printing of statistics: when the query was made, the size of the reply and so on. The default behaviour is to print the query statistics.

`+[no]qr`

Print `[do not print]` the query as it is sent. By default, the query is not printed.

`+[no]question`

Print `[do not print]` the question section of a query when an answer is returned. The default is to print the question section as a comment.

`+[no]answer`

Display `[do not display]` the answer section of a reply. The default is to display it.

`+[no]authority`

Display `[do not display]` the authority section of a reply. The default is to display it.

`+[no]additional`

Display `[do not display]` the additional section of a reply. The default is to display it.

`+[no]all`

Set or clear all display flags.

`+time=T`

Sets the timeout for a query to T seconds. The default time out is 5 seconds. An attempt to set T to less than 1 will result in a query timeout of 1 second being applied.

`+tries=T`

Sets the number of times to retry UDP queries to server to T instead of the default, 3. If T is less than or equal to zero, the number of retries is silently rounded up to 1.

`+ndots=D`

Set the number of dots that have to appear in name to D for it to be considered absolute. The default value is that defined using the `ndots` statement in `/etc/resolv.conf`, or 1 if no `ndots` statement is present. Names with fewer dots are interpreted as relative names and will be searched for in the domains listed in the `search` or `domain` directive in `/etc/resolv.conf`.

`+bufsize=B`

Set the UDP message buffer size advertised using EDNS0 to B bytes. The maximum and minimum sizes of this buffer are 65535 and 0 respectively. Values outside this range are rounded up or

down appropriately.

`+[no]multiline`

Print records like the SOA records in a verbose multi-line format with human-readable comments. The default is to print each record on a single line, to facilitate machine parsing of the dig output.

`+[no]fail`

Do not try the next server if you receive a SERVFAIL. The default is to not try the next server which is the reverse of normal stub resolver behaviour.

`+[no]besteffort`

Attempt to display the contents of messages which are malformed. The default is to not display malformed answers.

`+[no]dnssec`

Requests DNSSEC records be sent by setting the DNSSEC OK bit (DO) in the the OPT record in the additional section of the query.

#### MULTIPLE QUERIES

The BIND 9 implementation of dig supports specifying multiple queries on the command line (in addition to supporting the `-f` batch file option). Each of those queries can be supplied with its own set of flags, options and query options.

In this case, each query argument represent an individual query in the command-line syntax described above. Each consists of any of the standard options and flags, the name to be looked up, an optional query type and class and any query options that should be applied to that query.

A global set of query options, which should be applied to all queries, can also be supplied. These global query options must precede the first tuple of name, class, type, options, flags, and query options supplied on the command line. Any global query options (except the `+[no]cmd` option) can be overridden by a query-specific set of query options. For example:

```
dig +qr www.isc.org any -x 127.0.0.1 isc.org ns +noqr
```

shows how dig could be used from the command line to make three lookups: an ANY query for `www.isc.org`, a reverse lookup of `127.0.0.1` and a query for the NS records of `isc.org`. A global query option of `+qr` is applied, so that dig shows the initial query it made for each lookup. The final query has a local query option of `+noqr` which means that dig will not print the initial query when it looks up the NS records for `isc.org`.

FILES

`/etc/resolv.conf`

SEE ALSO

`host(1)`, `named(8)`, `dnssec-keygen(8)`, RFC1035.

BUGS

There are probably too many query options.

BIND9 Jun 30, 2000 DIG(1)