

NAME

delv – DNS lookup and validation utility

SYNOPSIS

delv [@server] [[-4] | [-6]] [-a *anchor-file*] [-b *address*] [-c *class*] [-d *level*] [-i] [-m] [-p *port#*]
 [-q *name*] [-t *type*] [-x *addr*] [*name*] [*type*] [*class*] [*queryopt...*]

delv [-h]

delv [-v]

delv [*queryopt...*] [*query...*]

DESCRIPTION

delv is a tool for sending DNS queries and validating the results, using the same internal resolver and validator logic as **named**.

delv will send to a specified name server all queries needed to fetch and validate the requested data; this includes the original requested query, subsequent queries to follow CNAME or DNAME chains, and queries for DNSKEY and DS records to establish a chain of trust for DNSSEC validation. It does not perform iterative resolution, but simulates the behavior of a name server configured for DNSSEC validating and forwarding.

By default, responses are validated using built-in DNSSEC trust anchor for the root zone ("."). Records returned by **delv** are either fully validated or were not signed. If validation fails, an explanation of the failure is included in the output; the validation process can be traced in detail. Because **delv** does not rely on an external server to carry out validation, it can be used to check the validity of DNS responses in environments where local name servers may not be trustworthy.

Unless it is told to query a specific name server, **delv** will try each of the servers listed in */etc/resolv.conf*. If no usable server addresses are found, **delv** will send queries to the localhost addresses (127.0.0.1 for IPv4, ::1 for IPv6).

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

SIMPLE USAGE

A typical invocation of **delv** looks like:

```
delv @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, **delv** resolves that name before querying that name server (note, however, that this initial lookup is *not* validated by DNSSEC).

If no *server* argument is provided, **delv** consults */etc/resolv.conf*; if an address is found there, it queries the name server at that address. If either of the -4 or -6 options are in use, then only addresses for the corresponding transport will be tried. If no usable addresses are found, **delv** will send queries to the localhost addresses (127.0.0.1 for IPv4, ::1 for IPv6).

name

is the domain name to be looked up.

type

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

OPTIONS

-a *anchor-file*

Specifies a file from which to read DNSSEC trust anchors. The default is `/etc/bind.keys`, which is included with BIND 9 and contains one or more trust anchors for the root zone (".").

Keys that do not match the root zone name are ignored. An alternate key name can be specified using the **+root=NAME** options.

Note: When reading the trust anchor file, **delv** treats **trust-anchorsinitial-key** and **static-key** entries identically. That is, even if a key is configured with **initial-key**, indicating that it is meant to be used only as an initializing key for RFC 5011 key maintenance, it is still treated by **delv** as if it had been configured as a **static-key**. **delv** does not consult the managed keys database maintained by **named**. This means that if either of the keys in `/etc/bind.keys` is revoked and rolled over, it will be necessary to update `/etc/bind.keys` to use DNSSEC validation in **delv**.

-b *address*

Sets the source IP address of the query to *address*. This must be a valid address on one of the host's network interfaces or "0.0.0.0" or ":::". An optional source port may be specified by appending "#<port>".

-c *class*

Sets the query class for the requested data. Currently, only class "IN" is supported in **delv** and any other value is ignored.

-d *level*

Set the systemwide debug level to **level**. The allowed range is from 0 to 99. The default is 0 (no debugging). Debugging traces from **delv** become more verbose as the debug level increases. See the **+mtrace**, **+rtrace**, and **+vtrace** options below for additional debugging details.

-h

Display the **delv** help usage output and exit.

-i

Insecure mode. This disables internal DNSSEC validation. (Note, however, this does not set the CD bit on upstream queries. If the server being queried is performing DNSSEC validation, then it will not return invalid data; this can cause **delv** to time out. When it is necessary to examine invalid data to debug a DNSSEC problem, use **dig +cd**.)

-m

Enables memory usage debugging.

-p *port#*

Specifies a destination port to use for queries instead of the standard DNS port number 53. This option would be used with a name server that has been configured to listen for queries on a non-standard port number.

-q *name*

Sets the query name to *name*. While the query name can be specified without using the **-q**, it is sometimes necessary to disambiguate names from types or classes (for example, when looking up the name "ns", which could be misinterpreted as the type NS, or "ch", which could be misinterpreted as class CH).

-t *type*

Sets the query type to *type*, which can be any valid query type supported in BIND 9 except for zone transfer types AXFR and IXFR. As with **-q**, this is useful to distinguish query name type or class when they are ambiguous. It is sometimes necessary to disambiguate names from types.

The default query type is "A", unless the **-x** option is supplied to indicate a reverse lookup, in which case it is "PTR".

- v
Print the **delv** version and exit.
- x *addr*
Performs a reverse lookup, mapping an addresses to a name. *addr* is an IPv4 address in dotted–decimal notation, or a colon–delimited IPv6 address. When –x is used, there is no need to provide the *name* or *type* arguments. **delv** automatically performs a lookup for a name like 11.12.13.10.in–addr.arpa and sets the query type to PTR. IPv6 addresses are looked up using nibble format under the IP6.ARPA domain.
- 4
Forces **delv** to only use IPv4.
- 6
Forces **delv** to only use IPv6.

QUERY OPTIONS

delv provides a number of query options which affect the way results are displayed, and in some cases the way lookups are performed.

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]*cdflag

Controls whether to set the CD (checking disabled) bit in queries sent by **delv**. This may be useful when troubleshooting DNSSEC problems from behind a validating resolver. A validating resolver will block invalid responses, making it difficult to retrieve them for analysis. Setting the CD flag on queries will cause the resolver to return invalid responses, which **delv** can then validate internally and report the errors in detail.

+*[no]*class

Controls whether to display the CLASS when printing a record. The default is to display the CLASS.

+*[no]*ttdl

Controls whether to display the TTL when printing a record. The default is to display the TTL.

+*[no]*rtrace

Toggle resolver fetch logging. This reports the name and type of each query sent by **delv** in the process of carrying out the resolution and validation process: this includes including the original query and all subsequent queries to follow CNAMEs and to establish a chain of trust for DNSSEC validation.

This is equivalent to setting the debug level to 1 in the "resolver" logging category. Setting the systemwide debug level to 1 using the –d option will product the same output (but will affect other logging categories as well).

+*[no]*mtrace

Toggle message logging. This produces a detailed dump of the responses received by **delv** in the process of carrying out the resolution and validation process.

This is equivalent to setting the debug level to 10 for the "packets" module of the "resolver" logging category. Setting the systemwide debug level to 10 using the –d option will produce the same output (but will affect other logging categories as well).

+*[no]*vtrace

Toggle validation logging. This shows the internal process of the validator as it determines whether an answer is validly signed, unsigned, or invalid.

This is equivalent to setting the debug level to 3 for the "validator" module of the "dnssec" logging

category. Setting the systemwide debug level to 3 using the **-d** option will produce the same output (but will affect other logging categories as well).

+`[no]short`

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

+`[no]comments`

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

+`[no]rrcomments`

Toggle the display of per-record comments in the output (for example, human-readable key information about DNSKEY records). The default is to print per-record comments.

+`[no]crypto`

Toggle the display of cryptographic fields in DNSSEC records. The contents of these field are unnecessary to debug most DNSSEC validation failures and removing them makes it easier to see the common failures. The default is to display the fields. When omitted they are replaced by the string "[omitted]" or in the DNSKEY case the key id is displayed as the replacement, e.g. "[key id = value]".

+`[no]trust`

Controls whether to display the trust level when printing a record. The default is to display the trust level.

+`[no]split[=W]`

Split long hex- or base64-formatted fields in resource records into chunks of *W* characters (where *W* is rounded up to the nearest multiple of 4). *+nosplit* or *+split=0* causes fields not to be split at all. The default is 56 characters, or 44 characters when multiline mode is active.

+`[no]all`

Set or clear the display options **+`[no]comments`**, **+`[no]rrcomments`**, and **+`[no]trust`** as a group.

+`[no]multiline`

Print long records (such as RRSIG, DNSKEY, and 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 **delv** output.

+`[no]dnssec`

Indicates whether to display RRSIG records in the **delv** output. The default is to do so. Note that (unlike in **dig**) this does *not* control whether to request DNSSEC records or whether to validate them. DNSSEC records are always requested, and validation will always occur unless suppressed by the use of **-i** or **+noroot**.

+`[no]root[=ROOT]`

Indicates whether to perform conventional DNSSEC validation, and if so, specifies the name of a trust anchor. The default is to validate using a trust anchor of "." (the root zone), for which there is a built-in key. If specifying a different trust anchor, then **-a** must be used to specify a file containing the key.

+`[no]tcp`

Controls whether to use TCP when sending queries. The default is to use UDP unless a truncated response has been received.

+`[no]unknownformat`

Print all RDATA in unknown RR type presentation format (RFC 3597). The default is to print RDATA for known types in the type's presentation format.

+`[no]yaml`

Print response data in YAML format.

FILES

/etc/bind.keys

`/etc/resolv.conf`

SEE ALSO

`dig(1)`, `named(8)`, RFC4034, RFC4035, RFC4431, RFC5074, RFC5155.

AUTHOR

Internet Systems Consortium, Inc.

COPYRIGHT

Copyright © 2014-2020 Internet Systems Consortium, Inc. ("ISC")