# dig DNS lookups

**START HERE**

`dig <name>`
Look up A records

`dig +noall +answer <name>`
Answer section only

`dig +short <name>`
Just the answer

`dig <name> ANY`
Ask for everything

`dig <name> MX`
Ask for a record type

`dig @<server> <name>`
Ask a specific server

`dig -x <ip>`
Reverse lookup

`dig +trace <name>`
Follow delegation from root

**Your machine** dig → `query` → **Resolver** @server or resolv.conf → `+trace skips it` → **Root / TLD** delegations → `refers to` → **Authoritative** the final answer

## CONCEPTS

dig talks to the DNS server directly and ignores /etc/hosts, so it can differ from what your applications resolve.

Options starting with + change a query; options starting with - change dig itself. Order is free: name, type and @server may come anywhere.

## BASIC QUERIES

`dig <name>`
Query A records

`dig <name> <type>`
Query a record type

`dig -t <type> <name>`
Same, with the type as a flag

`dig <name> AAAA`
IPv6 address

`dig <name> MX`
Mail servers

`dig <name> NS`
Name servers of a zone

`dig <name> TXT`
Text records (SPF, verification)

`dig <name> CNAME`
Alias target

`dig <name> SOA`
Zone start of authority and serial

## BASIC QUERIES (cont.)

`dig <name> SRV`
Service location records

`dig <name> CAA`
Certificate authority restrictions

`dig <name> ANY`
Request all types
*Many servers answer ANY minimally or refuse it, so do not trust it as a full listing.*

`dig <name1> <name2>`
Several lookups in one run

## CHOOSE THE SERVER

`dig @<server> <name>`
Ask this resolver or server

`dig @1.1.1.1 <name>`
Use a public resolver

`dig @<ns> <name> +norecurse`
Ask without recursion

`dig -p <port> @<server> <name>`
Non-standard port

`dig -4 <name>`
Use IPv4 transport only

`dig -6 <name>`
Use IPv6 transport only

`dig -b <addr> <name>`
Bind to a source address

`dig +tcp <name>`
Query over TCP

## CHOOSE THE SERVER (cont.)

`dig +tls @<server> <name>`
DNS over TLS

`dig +https @<server> <name>`
DNS over HTTPS

`dig +nsid @<server> <name>`
Ask the server to identify itself

## OUTPUT CONTROL

`dig +short <name>`
Print only the answer data

`dig +noall +answer <name>`
Answer section only, with TTLs

`dig +noall +answer +ttlunits <name>`
TTLs as 1h30m style

`dig +nocmd <name>`
Hide the command banner

`dig +nocomments <name>`
Hide section comments

`dig +nostats <name>`
Hide the timing footer

`dig +multiline <name> SOA`
Expand records over lines

`dig +short -x <ip>`
Reverse lookup, name only

## REVERSE AND TRACING

`dig -x <ip>`
Reverse (PTR) lookup

## REVERSE AND TRACING (cont.)

`dig +trace <name>`
Walk from the root servers down
*Uses its own iterative lookups, so it ignores your resolver's cache.*

`dig +nssearch <zone>`
Ask every authoritative server for the SOA

`dig NS <zone> +short`
List the zone's name servers

## DNSSEC AND EDNS

`dig +dnssec <name>`
Request signatures (RRSIG)

## TIMING AND RELIABILITY

`dig +timeout=<sec> <name>`
Per-try timeout (default 5)

`dig +tries=<n> <name>`
UDP attempts (default 3)

## BATCH AND CONFIG

`dig -f <file>`
Run lookups listed in a file
*Each line is written like a command line: name, type, options.*

`dig -v`
Print dig's version

## BATCH AND CONFIG (cont.)

**~/.digrc defaults**

```
# one line, read on every run (-r skips it)
+noall +answer
```

**Batch file**

```
# queries.txt
example.com A
example.com MX +short
-x 8.8.8.8
```

## READING THE OUTPUT

**Header flags**

| Item | Meaning |
|---|---|
| `status: NOERROR` | Query succeeded (answer may be empty) |
| `status: NXDOMAIN` | Name does not exist |
| `status: SERVFAIL` | Server could not answer |
| `status: REFUSED` | Server will not answer you |
| `flags: qr` | This is a response |
| `flags: aa` | Authoritative answer |
| `flags: rd ra` | Recursion asked / available |
| `flags: ad` | DNSSEC validated |
| `flags: tc` | Truncated, retry over TCP |

**Answer line**

| Field | Example |
|---|---|
| name | example.com. |
| TTL (seconds) | 300 |
| class | IN |
| type | A |
| data | 93.184.216.34 |

## READING THE OUTPUT (cont.)

NOERROR with an empty answer means the name exists but has no record of that type. NXDOMAIN means the name does not exist at all.

A falling TTL on repeated queries to a resolver shows the answer comes from its cache.

## RECIPES

**Compare resolvers**

```
for s in 1.1.1.1 8.8.8.8 9.9.9.9; do
  echo "$s: $(dig +short @$s <name>)"
done
```

**Check each authoritative server**

```
for ns in $(dig +short NS <zone>); do
  dig +short @$ns SOA <zone>
done
# differing serials mean zone replication lag
```

**Debug a flaky lookup**

```
dig +tries=1 +timeout=2 +norecurse @<server> <name>
dig +trace <name> | tail -20
```

**Bulk reverse lookups**

```
while read ip; do
  echo "$ip $(dig +short -x $ip)"
done < ips.txt
```