

Uplivra is in beta and online purchasing is paused. **Apply to test it:** testers get a license on request and 10% off when purchasing opens.

INSTALL GUIDE · IT TEAMS AND MSPS

DNS from the command line

Every uplivra dns command, with examples, for zones and records, CSV import and export, upstream and forwarder checks, lookups and dig, and scripts in bash or PowerShell.

Uplivra Technologies LLC · Guide for Uplivra 26.10 · Updated 11 October 2026 · Latest version:
<https://uplivra.com/guides/dns-command-line.html>

Where to run it

What	Where	Rights
Zones and records, import and export	The Uplivra server	root, the uplivra account on an appliance, or a member of the uplivra group
status , upstreams , forwarders , compare	The Uplivra server (every DNS server), or a DNS appliance (just itself)	The same
lookup , dig , test	Any computer with Uplivra, Windows too	None

On an appliance, sign in as uplivra and type the commands as they are here, without sudo . Uplivra asks for the rights a command needs by itself.

uplivra dns help lists every command, and uplivra dns help import shows one. Every change goes into the audit log with who made it (for example cli:uplivra).

Zones

```
uplivra dns zones
uplivra dns zone add office.example.com -ttl 300 -auto-dhcp
uplivra dns zone remove lab.example.com
```

- **-auto-dhcp** adds names from Uplivra's DHCP leases, and **-auto-ipam** adds names from IP address reservations. See [Automatic names](#).
- **zone remove** asks first. Add **-yes** in a script.

Records

```
uplivra dns records office.example.com
uplivra dns add office.example.com printer A 192.168.10.20
uplivra dns add office.example.com @ MX mail.example.com. -priority 10
uplivra dns add office.example.com _sip._tcp SRV pbx -priority 10 -weight 5 -port 5060
uplivra dns add office.example.com @ TXT "v=spf1 mx -all"
uplivra dns set office.example.com www CNAME web1
uplivra dns edit office.example.com 42 -value 192.168.10.21
uplivra dns delete office.example.com 42
uplivra dns delete office.example.com fax A
uplivra dns find 192.168.10.20
```

- **records** shows each record's number in the **#** column. **edit** and **delete** use that number.
- **add** leaves out an exact copy of a record that is already there. With **-if-missing**, it adds nothing when the name already has a record of that type, whatever its value.
- **set** makes a name and type exactly one value, replacing what was there. You can run it again and again in a script with the same result.
- **delete** takes a record number, or a name and type, with or without the value.
- **find** looks through every zone's names and values, automatic names included.
- A change that breaks a rule is refused, and nothing is saved. For example, a CNAME must be the only record at its name.

DNS servers answer with a change within a minute.

CSV import and export

```
uplivra dns export office.example.com -csv > office.csv
uplivra dns import office.example.com office.csv -dry-run
uplivra dns import office.example.com office.csv
uplivra dns import office.example.com office.csv -replace
uplivra dns export -all > all-zones.csv
uplivra dns import -all all-zones.csv -dry-run
```

- **-dry-run** shows each record that would be added (+) or removed (-), and changes nothing. Use it first.
- **-replace** makes the zone exactly what the file has. It asks first; add **-yes** in a script.
- **import** also reads BIND-style zone files, such as Windows' `dnscmd /zoneexport` and most DNS hosts' exports. **export** without **-csv** writes one.
- **-** in place of a file name reads standard input: `cat office.csv | uplivra dns import office.example.com -`.

The CSV columns

Column	Needed	What
zone	For -all only	The zone the record belongs to
name	Yes	@ for the zone itself, <code>printer</code> , <code>_sip._tcp</code>
type	Yes	A, AAAA, CNAME, MX, TXT, SRV, PTR, NS, CAA
value	Yes	The address, host name or text
ttl	No	Seconds; empty or 0 uses the zone's
priority	MX, SRV	The lower number is tried first
weight	SRV	
port	SRV	
source	Export only	<code>manual</code> , or <code>dhcp</code> , <code>ipam</code> , <code>ddns</code> for automatic names

- The first line names the columns, in any order and any case. A file without that line is read as name, type, value, ttl, priority, weight, port.
- Files saved by Excel work: "CSV UTF-8", and files split by semicolons (Excel outside the US).
- **Left out on import, and listed:**
 - exact copies of records already there;
 - automatic names (the `source` column);
 - lines with a mistake, each with the reason.
- **-automatic** on export adds the automatic names, marked in the `source` column, for a full list. Importing that file leaves them out again, because Uplivra makes them itself.
- **Formulas:** a TXT value starting with `=`, `+`, `-` or `@` is written with a leading `'`, so Excel doesn't treat it as a formula. Import takes the `'` off again.

Many changes at once

Put one change on each line, using the same words as above without `uplivra dns`:

```
# changes.txt
add office.example.com ns2 A 192.168.10.3
set office.example.com www CNAME web2
add office.example.com @ TXT "google-site-verification=abc123"
delete office.example.com oldprinter A
```

```
uplivra dns batch changes.txt -dry-run
uplivra dns batch changes.txt
```

A batch is all or nothing. If one line can't be done, for example a record that isn't there, it names the line and changes nothing.

The DNS servers

```
uplivra dns status
uplivra dns upstreams -check
uplivra dns forwarders -check -ad
uplivra dns compare office.example.com
```

- **status:** each DNS server: on or off, its listening addresses, queries, the share answered from its cache, its upstreams' health, blocks (with DNS Security) and any problem, such as an address it can't listen on.
- **upstreams -check:** asks each upstream resolver now, the way the DNS server does. That means DNS over TLS on port 853 with the certificate's name checked, or plain DNS on port 53.
- It says whether it answered and how fast.
- It says whether the answers are DNSSEC-checked, and when a certificate is about to expire.
- **-ask NAME** looks up another name than example.com.
- **forwarders -check:** asks each conditional forwarder for its zone's SOA. **-ad** also checks the zone lists Active Directory domain controllers (`_ldap._tcp.dc._msdcs`).
- **compare:** the zone's SOA serial on every Uplivra DNS server and every secondary. A lower serial means an older copy.

On the Uplivra server, these list every DNS server. The checks are made from the server itself. To check from a DNS server's own network, run the same command on that appliance.

Lookups and tests

```
uplivra dns lookup printer.office.example.com
uplivra dns lookup 192.168.10.20
uplivra dns lookup example.com -type MX -server 192.168.10.2
uplivra dns test badsite.example -server 192.168.10.2
uplivra dns dig @192.168.10.2 example.com MX
uplivra dns dig example.com +short
uplivra dns dig @9.9.9.9 example.com +tls +dnssec
uplivra dns dig example.com +trace
uplivra dns dig -x 192.168.10.20
```

- **lookup** answers in plain words and follows CNAMEs. Give it an IP address to look up its name.
- **test** asks one DNS server and explains its answer, including why a name was blocked: DNS Security, or Quad9's block.
- **dig** prints dig's usual output.
- Options: **+short**, **+tcp**, **+tls** (DNS over TLS), **+tls-host=NAME**, **+dnssec**, **+norecurse**, **+trace**, **+time=N**, **-x**, **-p PORT**, **-t TYPE**.
- **+trace** follows the name down from the root servers.
- Without **-server** (or **@** for dig), they ask this computer's own DNS server. On Windows, give one.

For scripts

Option	What
<code>-json</code>	JSON instead of text, on every command
<code>-q</code>	Print nothing; use the exit code
<code>-dry-run</code>	Show what would change; change nothing
<code>-yes</code>	Don't ask before replacing or removing
<code>-timeout 4s</code>	How long to wait for each DNS answer
<code>-</code>	Read standard input instead of a file

Exit code	Means
0	Done
1	Failed: the message says why
2	Wrong use: an unknown command or option
3	Not found, or a check failed: a name that doesn't exist, an upstream or forwarder that didn't answer, <code>find</code> found nothing

Results go to standard output and messages to standard error, so `>` and `|` carry only the results.

Bash

```
#!/bin/bash
# Every night: keep a CSV copy of every zone.
uplivra dns export -all > /var/backups/uplivra/dns-$(date +%F).csv

# Alert when an upstream fails.
if ! uplivra dns upstreams -check -q; then
    echo "an Uplivra DNS upstream is failing" | mail -s "DNS" it@example.com
fi

# Add the printers from a list, once each.
while read name addr; do
    uplivra dns add office.example.com "$name" A "$addr" -if-missing -q
done < printers.txt

# Names that point at an address you are retiring.
uplivra dns find 192.168.10.50 -json | jq -r '.[].name'
```

PowerShell

Over SSH to the server, or with `uplivra.exe` for lookups on Windows:

```
# Which upstreams failed, as objects
ssh uplivra@uplivra.example.com "uplivra dns upstreams -check -json" |
    ConvertFrom-Json | ForEach-Object { $_.checks } | Where-Object { -not $_.ok }

# Does the name resolve through the office DNS server?
uplivra.exe dns lookup intranet.office.example.com -server 192.168.10.2 -q
if ($LASTEXITCODE -eq 3) { Write-Warning "intranet doesn't resolve" }
```

To manage records from a computer without SSH to the server, use the management API (`/api/v1/dns/...`) with a token. See the [API reference](#).