NAME

resolvectl, resolvconf − Resolve domain names, IPV4 and IPv6 addresses, DNS resource records, and services; introspect and reconfigure the DNS resolver

SYNOPSIS

resolvectl [OPTIONS...] {COMMAND} [NAME...]

DESCRIPTION

resolvectl may be used to resolve domain names, IPv4 and IPv6 addresses, DNS resource records and services with the **systemd-resolved.service**(8) resolver service. By default, the specified list of parameters will be resolved as hostnames, retrieving their IPv4 and IPv6 addresses. If the parameters specified are formatted as IPv4 or IPv6 operation the reverse operation is done, and a hostname is retrieved for the specified addresses.

The program's output contains information about the protocol used for the look−up and on which network interface the data was discovered. It also contains information on whether the information could be authenticated. All data for which local DNSSEC validation succeeds is considered authenticated. Moreover all data originating from local, trusted sources is also reported authenticated, including resolution of the local host name, the "localhost" host name or all data from /etc/hosts.

OPTIONS

−4, **−6**

By default, when resolving a hostname, both IPv4 and IPv6 addresses are acquired. By specifying **−4** only IPv4 addresses are requested, by specifying **−6** only IPv6 addresses are requested.

−i *INTERFACE*, **−−interface=***INTERFACE*

Specifies the network interface to execute the query on. This may either be specified as numeric interface index or as network interface string (e.g. "en0"). Note that this option has no effect if system−wide DNS configuration (as configured in /etc/resolv.conf or /etc/systemd/resolve.conf) in place of per−link configuration is used.

−p *PROTOCOL*, **−−protocol=***PROTOCOL*

Specifies the network protocol for the query. May be one of "dns" (i.e. classic unicast DNS), "llmnr" (**Link−Local Multicast Name Resolution**[1]), "llmnr−ipv4", "llmnr−ipv6" (LLMNR via the indicated underlying IP protocols), "mdns" (**Multicast DNS**[2]), "mdns−ipv4", "mdns−ipv6" (MDNS via the indicated underlying IP protocols). By default the lookup is done via all protocols suitable for the lookup. If used, limits the set of protocols that may be used. Use this option multiple times to enable resolving via multiple protocols at the same time. The setting "llmnr" is identical to specifying this switch once with "llmnr−ipv4" and once via "llmnr−ipv6". Note that this option does not force the service to resolve the operation with the specified protocol, as that might require a suitable network interface and configuration. The special value "help" may be used to list known values.

−t *TYPE*, **−−type=***TYPE*, **−c** *CLASS*, **−−class=***CLASS*

Specifies the DNS resource record type (e.g. A, AAAA, MX, ...) and class (e.g. IN, ANY, ...) to look up. If these options are used a DNS resource record set matching the specified class and type is requested. The class defaults to IN if only a type is specified. The special value "help" may be used to list known values.

−−service−address=*BOOL*

Takes a boolean parameter. If true (the default), when doing a service lookup with **−−service** the hostnames contained in the SRV resource records are resolved as well.

−−service−txt=*BOOL*

Takes a boolean parameter. If true (the default), when doing a DNS−SD service lookup with **−−service** the TXT service metadata record is resolved as well.

−−cname=*BOOL*

Takes a boolean parameter. If true (the default), DNS CNAME or DNAME redirections are followed. Otherwise, if a CNAME or DNAME record is encountered while resolving, an error is returned.

−−search=*BOOL*

Takes a boolean parameter. If true (the default), any specified single−label hostnames will be searched in the domains configured in the search domain list, if it is non−empty. Otherwise, the search domain logic is disabled.

−−raw[=payload|packet]
Dump the answer as binary data. If there is no argument or if the argument is "payload", the payload of the packet is exported. If the argument is "packet", the whole packet is dumped in wire format, prefixed by length specified as a little−endian 64−bit number. This format allows multiple packets to be dumped and unambiguously parsed.

−−legend=*BOOL*
Takes a boolean parameter. If true (the default), column headers and meta information about the query response are shown. Otherwise, this output is suppressed.

−h, **−−help**
Print a short help text and exit.

−−version
Print a short version string and exit.

−−no−pager
Do not pipe output into a pager.

COMMANDS

query *HOSTNAME*|*ADDRESS*...
Resolve domain names, IPv4 and IPv6 addresses.

service [[*NAME*] *TYPE*] *DOMAIN*
Resolve **DNS−SD**[3] and **SRV**[4] services, depending on the specified list of parameters. If three parameters are passed the first is assumed to be the DNS−SD service name, the second the SRV service type, and the third the domain to search in. In this case a full DNS−SD style SRV and TXT lookup is executed. If only two parameters are specified, the first is assumed to be the SRV service type, and the second the domain to look in. In this case no TXT RR is requested. Finally, if only one parameter is specified, it is assumed to be a domain name, that is already prefixed with an SRV type, and an SRV lookup is done (no TXT).

openpgp *EMAIL@DOMAIN*...
Query PGP keys stored as **OPENPGPKEY**[5] resource records. Specified e−mail addresses are converted to the corresponding DNS domain name, and any OPENPGPKEY keys are printed.

tlsa [*FAMILY*] *DOMAIN*[**:***PORT*]...
Query TLS public keys stored as **TLSA**[6] resource records. A query will be performed for each of the specified names prefixed with the port and family ("*_port._family.domain*"). The port number may be specified after a colon (":"), otherwise **443** will be used by default. The family may be specified as the first argument, otherwise **tcp** will be used.

status [*LINK*...]
Shows the global and per−link DNS settings in currently in effect. If no command is specified, this is the implied default.

statistics
Shows general resolver statistics, including information whether DNSSEC is enabled and available, as well as resolution and validation statistics.

reset−statistics
Resets the statistics counters shown in **statistics** to zero. This operation requires root privileges.

flush−caches
Flushes all DNS resource record caches the service maintains locally. This is mostly equivalent to sending the **SIGUSR2** to the **systemd−resolved** service.

reset−server−features
Flushes all feature level information the resolver learnt about specific servers, and ensures that the

server feature probing logic is started from the beginning with the next look−up request. This is mostly equivalent to sending the **SIGRTMIN+1** to the **systemd−resolved** service.

dns [*LINK* [*SERVER*...]], **domain** [*LINK* [*DOMAIN*...]], **default−route** [*LINK* [*BOOL*...]], **llmnr** [*LINK* [*MODE*]], **mdns** [*LINK* [*MODE*]], **dnssec** [*LINK* [*MODE*]], **dnsovertls** [*LINK* [*MODE*]], **nta** [*LINK* [*DOMAIN*...]]

Get/set per−interface DNS configuration. These commands may be used to configure various DNS settings for network interfaces that aren't managed by **systemd-networkd.service**(8). (These commands will fail when used on interfaces that are managed by **systemd−networkd**, please configure their DNS settings directly inside the .network files instead.) These commands may be used to inform **systemd−resolved** about per−interface DNS configuration determined through external means. The **dns** command expects IPv4 or IPv6 address specifications of DNS servers to use. The **domain** command expects valid DNS domains, possibly prefixed with "~", and configures a per−interface search or route−only domain. The **default−route** command expects a boolean paremeter, and configures whether the link may be used as default route for DNS lookups, i.e. if it is suitable for lookups on domains no other link explicitly is configured for. The **llmnr**, **mdns**, **dnssec** and **dnsovertls** commands may be used to configure the per−interface LLMNR, MulticastDNS, DNSSEC and DNSOverTLS settings. Finally, **nta** command may be used to configure additional per−interface DNSSEC NTA domains.

Options **dns**, **domain** and **nta** can take a single empty string argument to clear their respective value lists.

For details about these settings, their possible values and their effect, see the corresponding options in **systemd.network**(5).

revert *LINK*

Revert the per−interface DNS configuration. If the DNS configuration is reverted all per−interface DNS setting are reset to their defaults, undoing all effects of **dns**, **domain**, **default−route**, **llmnr**, **mdns**, **dnssec**, **dnsovertls**, **nta**. Note that when a network interface disappears all configuration is lost automatically, an explicit reverting is not necessary in that case.

COMPATIBILITY WITH RESOLVCONF(8)

resolvectl is a multi−call binary. When invoked as "resolvconf" (generally achieved by means of a symbolic link of this name to the **resolvectl** binary) it is run in a limited **resolvconf**(8) compatibility mode. It accepts mostly the same arguments and pushes all data into **systemd-resolved.service**(8), similar to how **dns** and **domain** commands operate. Note that **systemd−resolved.service** is the only supported backend, which is different from other implementations of this command. Note that not all operations supported by other implementations are supported natively. Specifically:

−a

Registers per−interface DNS configuration data with **systemd−resolved**. Expects a network interface name as only command line argument. Reads **resolv.conf**(5) compatible DNS configuration data from its standard input. Relevant fields are "nameserver" and "domain"/"search". This command is mostly identical to invoking **resolvectl** with a combination of **dns** and **domain** commands.

−d

Unregisters per−interface DNS configuration data with **systemd−resolved**. This command is mostly identical to invoking **resolvectl revert**.

−f

When specified **−a** and **−d** will not complain about missing network interfaces and will silently execute no operation in that case.

−x

This switch for "exclusive" operation is supported only partially. It is mapped to an additional configured search domain of "~." — i.e. ensures that DNS traffic is preferably routed to the DNS servers on this interface, unless there are other, more specific domains configured on other interfaces.

−m, **−p**
These switches are not supported and are silently ignored.

−u, **−I**, **−i**, **−l**, **−R**, **−r**, **−v**, **−V**, **−−enable−updates**, **−−disable−updates**, **−−are−updates−enabled**
These switches are not supported and the command will fail if used.

See **resolvconf**(8) for details on this command line options.

EXAMPLES

Example 1. Retrieve the addresses of the "www.0pointer.net" domain

```
$ resolvectl query www.0pointer.net
www.0pointer.net: 2a01:238:43ed:c300:10c3:bcf3:3266:da74
                  85.214.157.71

−− Information acquired via protocol DNS in 611.6ms.
−− Data is authenticated: no
```

Example 2. Retrieve the domain of the "85.214.157.71" IP address

```
$ resolvectl query 85.214.157.71
85.214.157.71: gardel.0pointer.net

−− Information acquired via protocol DNS in 1.2997s.
−− Data is authenticated: no
```

Example 3. Retrieve the MX record of the "yahoo.com" domain

```
$ resolvectl −−legend=no −t MX query yahoo.com
yahoo.com. IN MX    1 mta7.am0.yahoodns.net
yahoo.com. IN MX    1 mta6.am0.yahoodns.net
yahoo.com. IN MX    1 mta5.am0.yahoodns.net
```

Example 4. Resolve an SRV service

```
$ resolvectl service _xmpp−server._tcp gmail.com
_xmpp−server._tcp/gmail.com: alt1.xmpp−server.l.google.com:5269 [priority=20, weight=0]
                             173.194.210.125
                             alt4.xmpp−server.l.google.com:5269 [priority=20, weight=0]
                             173.194.65.125
                             ...
```

Example 5. Retrieve a PGP key

```
$ resolvectl openpgp zbyszek@fedoraproject.org
d08ee310438ca124a6149ea5cc21b6313b390dce485576eff96f8722._openpgpkey.fedoraproject.org. IN OPENPGPKEY
    mQINBFBHPMsBEACeInGYJCb+7TurKfb6wGyTottCDtiSJB310i37/6ZYoeIay/5soJjlMyf
    MFQ9T2XNT/0LM6gTa0MpC1st9LnzYTMsT6tzRly1D1UbVI6xw0g0vE5y2Cjk3xUwAynCsSs
    ...
```

Example 6. Retrieve a TLS key ("tcp" and ":443" could be skipped)

```
$ resolvectl tlsa tcp fedoraproject.org:443
_443._tcp.fedoraproject.org IN TLSA 0 0 1 19400be5b7a31fb733917700789d2f0a2471c0c9d506c0e504c06c16d7cb17c0
    −− Cert. usage: CA constraint
    −− Selector: Full Certificate
    −− Matching type: SHA−256
```

SEE ALSO

systemd(1), **systemd-resolved.service**(8), **systemd.dnssd**(5), **systemd-networkd.service**(8), **resolvconf**(8)

NOTES

1. Link-Local Multicast Name Resolution
   https://tools.ietf.org/html/rfc4795
2. Multicast DNS
   https://www.ietf.org/rfc/rfc6762.txt
3. DNS-SD
   https://tools.ietf.org/html/rfc6763
4. SRV
   https://tools.ietf.org/html/rfc2782
5. OPENPGPKEY
   https://tools.ietf.org/html/rfc7929
6. TLSA
   https://tools.ietf.org/html/rfc6698