

**NAME**

resolv.conf - resolver configuration file

**SYNOPSIS**

/etc/resolv.conf

**DESCRIPTION**

The \_\_\_\_\_ is a set of routines in the C library that provide access to the Internet Domain Name System (DNS). The resolver configuration file contains information that is read by the resolver routines the first time they are invoked by a process. The file is designed to be human readable and contains a list of keywords with values that provide various types of resolver information.

If this file doesn't exist the only name server to be queried will be on the local machine; the domain name is determined from the host name and the domain search path is constructed from the domain name.

The different configuration options are:

**nameserver** Name server IP address

Internet address (in dot notation) of a name server that the resolver should query. Up to MAXNS (currently 3, see \_\_\_\_\_) name servers may be listed, one per keyword. If there are multiple servers, the resolver library queries them in the order listed. If no nameserver entries are present, the default is to use the name server on the local machine. (The algorithm used is to try a name server, and if the query times out, try the next, until out of name servers, then repeat trying all the name servers until a maximum number of retries are made.)

**domain** Local domain name.

Most queries for names within this domain can use short names relative to the local domain. If no domain entry is present, the domain is determined from the local host name returned by gethostname(3); the domain part is taken to be everything after the first '.'. Finally, if the host name does not contain a domain part, the root domain is assumed.

**search** Search list for host-name lookup.

The search list is normally determined from the local domain name; by default, it contains only the local domain name. This may be changed by listing the desired domain search path following the \_\_\_\_\_ keyword with spaces or tabs separating the names. Resolver queries having fewer than \_\_\_\_\_ dots (default is 1) in them will be attempted using each component of the search path in turn until a match is found. For environments with multiple subdomains please read options ndots:\_ below to avoid man-in-the-middle attacks and unnecessary traffic for the root-dns-servers. Note that this process may be slow and will generate a lot of network traffic if the servers for the listed domains are

not local, and that queries will time out if no server is available for one of the domains.

The search list is currently limited to six domains with a total of 256 characters.

#### sortlist

Sortlist allows addresses returned by `gethostbyname(3)` to be sorted. A sortlist is specified by IP address netmask pairs. The netmask is optional and defaults to the natural netmask of the net. The IP address and optional network pairs are separated by slashes. Up to 10 pairs may be specified. E.g.,  
 sortlist 130.155.160.0/255.255.240.0 130.155.0.0

#### options

Options allows certain internal resolver variables to be modified. The syntax is

```
options _____
```

where \_\_\_\_\_ is one of the following:

debug sets RES\_DEBUG in \_\_\_\_\_.

ndots:\_\_\_\_\_

sets a threshold for the number of dots which must appear in a name given to `res_query(3)` (see `resolver(3)`) before an \_\_\_\_\_ will be made. The default for \_\_\_\_\_ is 1, meaning that if there are any dots in a name, the name will be tried first as an absolute name before any \_\_\_\_\_ elements are appended to it.

timeout:\_\_\_\_\_

sets the amount of time the resolver will wait for a response from a remote name server before retrying the query via a different name server. Measured in seconds, the default is RES\_TIMEOUT (currently 5, see \_\_\_\_\_).

attempts:\_\_\_\_\_

sets the number of times the resolver will send a query to its name servers before giving up and returning an error to the calling application. The default is RES\_DFLRETRY (currently 2, see \_\_\_\_\_).

rotate sets RES\_ROTATE in \_\_\_\_\_, which causes round robin selection of nameservers from among those listed. This has the effect of spreading the query load among all listed servers, rather than having all clients try the first listed server first every time.

no-check-names

sets RES\_NOCHECKNAME in \_\_\_\_\_, which disables the modern BIND checking of incoming host names and mail names for invalid characters such as underscore (\_), non-ASCII, or control characters.

`inet6` sets `RES_USE_INET6` in \_\_\_\_\_. This has the effect of trying a AAAA query before an A query inside the `gethostbyname(3)` function, and of mapping IPv4 responses in IPv6 "tunneled form" if no AAAA records are found but an A record set exists.

Some programs behave strangely when this option is turned on.

The \_\_\_\_\_ and \_\_\_\_\_ keywords are mutually exclusive. If more than one instance of these keywords is present, the last instance wins.

The \_\_\_\_\_ keyword of a system's \_\_\_\_\_ file can be overridden on a per-process basis by setting the environment variable `LOCALDOMAIN` to a space-separated list of search domains.

The \_\_\_\_\_ keyword of a system's \_\_\_\_\_ file can be amended on a per-process basis by setting the environment variable `RES_OPTIONS` to a space-separated list of resolver options as explained above under options.

The keyword and value must appear on a single line, and the keyword (e.g., `nameserver`) must start the line. The value follows the keyword, separated by white space.

## FILES

\_\_\_\_\_, \_\_\_\_\_

## SEE ALSO

`gethostbyname(3)`, `resolver(3)`, `hostname(7)`, `named(8)`  
Name Server Operations Guide for BIND

## COLOPHON

This page is part of release 2.77 of the Linux \_\_\_\_\_ project. A description of the project, and information about reporting bugs, can be found at <http://www.kernel.org/doc/man-pages/>.

4th Berkeley Distribution

2004-10-31

RESOLV.CONF(5)

From:

<https://wiki.inf.ufpr.br/maziero/> - **Prof. Carlos Maziero**

Permanent link:

<https://wiki.inf.ufpr.br/maziero/doku.php?id=espec:man-resolv.conf>

Last update: **2008/06/11 15:55**

