NAME

resolver - resolver configuration file

SYNOPSIS

resolv.conf

DESCRIPTION

The resolver(3) is a set of routines in the C library which provide access to the Internet Domain Name System. 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.

On a normally configured system, setting this file manually should not be necessary. The only name server(s) to be queried will be on the local machine or automatically configured using DHCP or a similar mechanism, 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 IPv4 or IPv6 address of a name server that the resolver should query. Up to MAXNS (currently 3) 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 **search** keyword with spaces or tabs separating the names. Most resolver queries will be attempted using each component of the search path in turn until a match is found. 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 to be sorted. A sortlist is specified by IP address netmask pairs. If the netmask is not specified, it defaults to the historical Class A/B/C netmask of the net; this usage is deprecated. The IP address and network pairs are separated by slashes. Up to 10 pairs may be specified. E.g.,

```
sortlist 10.9.1.0/255.255.240.0 10.9.0.0/255.255.0.0
```

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

options *option* ...

where **option** is one of the following:

debug sets RES_DEBUG in _res.options.

usevc sets RES_USEVC to use TCP instead of UDP for queries.

ndots:*n* sets a threshold for the number of dots which must appear in a name given to **res_query**() (see resolver(3)) before an *initial absolute query* will be made. The default for *n* is "1", meaning that if there are any dots in a name, the name will be tried first as an absolute name before any *search list* elements are appended to it.

timeout:*n* sets the initial amount of time the resolver will wait for a response from a remote name server before retrying the query via a different name server. The resolver may wait longer during subsequent retries of the current query since an exponential back-off is applied to the timeout value. Measured in seconds, the default is RES_TIMEOUT, the allowed maximum is RES_MAXRETRANS (see <*resolv.h*>).

attempts:*n* sets the number of times the resolver will send a query to each of its name servers before giving up and returning an error to the calling application. The default is RES_DFLRETRY, the allowed maximum is RES_MAXRETRY (see <*resolv.h*>).

no_tld_query tells the resolver not to attempt to resolve a top level domain name, that is, a name that contains no dots. Use of this option does not prevent the resolver from obeying the standard **domain** and **search** rules with the given name.

reload-period:*n*
The resolver checks the modification time of */etc/resolv.conf* every *n* seconds. If */etc/resolv.conf* has changed, it is automatically reloaded. The default for *n* is two seconds. Setting it to zero disables the file check.

Options may also be specified as a space or tab separated list using the RES_OPTIONS environment variable.

The **domain** and **search** keywords are mutually exclusive. If more than one instance of these keywords is present, the last instance will override.

The keyword and value must appear on a single line, and the keyword (for example, **nameserver**) must start the line. The value follows the keyword, separated by white space.

FILES

/etc/resolv.conf The file **resolv.conf** resides in */etc*.

EXAMPLES

A basic resolv.conf file could be in the following form.

```
# The domain directive is only necessary, if your local
# router advertises something like localdomain and you have
# set up your hostnames via an external domain.
domain localdomain.tld

# In case you a running a local dns server or caching name server
# like local-unbound(8) for example.
nameserver 127.0.0.1

# IP address of the local or ISP name service
nameserver 192.168.2.1

# Fallback nameservers, in this case these from Google.
nameserver 8.8.8.8
nameserver 8.8.4.4

# Attach an OPT pseudo-RR for the EDNS0 extension,
# as specified in RFC 2671.
options edns0
```

SEE ALSO

gethostbyname(3), resolver(3), hostname(7), resolvconf(8)

HISTORY

The **resolv.conf** file format appeared in 4.3BSD.