

NAME

`resolvconf` - manage nameserver information

SYNOPSIS

`cat FILE | resolvconf -a IFACE.PROG`

`resolvconf -d IFACE.PROG`

`resolvconf -u`

`resolvconf --enable-updates | --disable-updates | --updates-are-enabled`

DESCRIPTION

The `resolvconf` package comprises a simple database for run-time nameserver information and a simple framework for notifying applications of changes in that information. `Resolvconf` thus sets itself up as the intermediary between programs that supply nameserver information and applications that use that information.

Information is added to or removed from the database using the `resolvconf` program. See the **OPTIONS** section below for a discussion of the available options.

SUPPLIERS OF NAMESERVER INFORMATION

Normally the `resolvconf` program is run only by network interface configuration programs such as `ifup(8)`, `ifdown`, `NetworkManager(8)`, `dhclient(8)`, and `pppd(8)`; and by local nameservers such as `dnsmasq(8)`. These programs obtain nameserver information from some source and push it to `resolvconf`.

dhclient

The `dhclient` program, for example, may receive nameserver addresses and domain search list information during its negotiation with the DHCP server; if so, its hook script `/etc/dhcp/dhclient-enter-hooks.d/resolvconf` pushes this information to `resolvconf`.

ifup

The `ifup` program can be used to configure network interfaces according to settings in `/etc/network/interfaces`. To make `ifup` push nameserver information to `resolvconf` when it configures an interface the administrator must add `dns-` option lines to the relevant `iface` stanza in `interfaces(5)`. The following option names are accepted: `dns-nameserver`, `dns-search`, and `dns-sortlist`.

To add a nameserver IP address, add an option line consisting of `dns-nameserver` and the address. To add multiple nameserver addresses, include multiple such `dns-nameserver` lines.

```
dns-nameserver 192.168.1.254
dns-nameserver 8.8.8.8
```

To add search domain names, add a line beginning with **dns-search**.

```
dns-search foo.org bar.com
```

The **dns-nameservers** option is also accepted and, unlike **dns-nameserver**, can be given multiple arguments, separated by spaces.

The **dns-domain** option is deprecated in favor of **dns-search**.

The resulting stanza might look like the following example.

```
iface eth0 inet static
    address 192.168.1.3
    netmask 255.255.255.0
    gateway 192.168.1.1
    dns-nameserver 192.168.1.254
    dns-nameserver 8.8.8.8
    dns-search foo.org bar.com
```

N.B.: On a machine where **resolvconf** has just been or is about to be installed and which previously relied on a static */etc/resolv.conf* file,

- the nameserver information in that static file, (which is to say the information on **nameserver**, **domain**, **search** and **sortlist** lines) should be migrated to the appropriate **iface** stanza(s) in */etc/network/interfaces(5)* as just described;
- options (which is to say, any **options** lines) should be migrated to */etc/resolvconf/resolv.conf.d/base*.

Command line

The administrator *can* run **resolvconf** from the command line to add or delete nameserver information, but this is not normally necessary or advisable.

CONSUMERS OF NAMESERVER INFORMATION

Nameserver information provided to **resolvconf** is stored for use by subscribers to **resolvconf**'s notification service. Subscriber packages that need to know when nameserver information has changed should install a script in */etc/resolvconf/update.d/* (or in */etc/resolvconf/update-libc.d/*: see below). For example, DNS caches such as [dnsmasq\(8\)](#) and [pdnsd\(8\)](#) subscribe to the notification service so that they know whither to forward queries. Client hook scripts will find the files containing nameserver information in the current directory.

libc

The most important software package that subscribes to the notification service is the GNU C Library [resolver\(3\)](#). This library is used by many applications that need to resolve domain names. When nameserver information is updated, the script `/etc/resolvconf/update.d/libc` generates a new version of the resolver configuration file, `/etc/resolvconf/run/resolv.conf`, as described below. If the new version of the file differs from the previously generated one then the hook scripts found in `/etc/resolvconf/update-libc.d/` are executed.

The dynamically generated resolver configuration file always starts with the contents of `/etc/resolvconf/resolv.conf.d/head` and ends with the contents of `/etc/resolvconf/resolv.conf.d/tail`. Between *head* and *tail* the *libc* script inserts dynamic nameserver information compiled from, first, information provided for configured interfaces; second, static information from `/etc/resolvconf/resolv.conf.d/base`. Specifically, it writes:

- 1)
 - up to three **nameserver** lines, ordered according to `/etc/resolvconf/interface-order`, possibly fewer if one of the addresses is a loopback address and the **TRUNCATE_NAMESERVER_LIST_AFTER_LOOPBACK_ADDRESS** environment variable is affirmatively set, as discussed in the **ENVIRONMENT VARIABLES** section;
- 2)
 - up to one **search** line containing the combined domain search list from all "domain" and "search" input lines, also ordered according to [interface-order\(5\)](#);
- 3)
 - all other non-comment input lines.

To make the resolver use this dynamically generated resolver configuration file the administrator should ensure that `/etc/resolv.conf` is a symbolic link to `/etc/resolvconf/run/resolv.conf`. This link is normally created on installation of the resolvconf package. The link is never modified by the **resolvconf** program itself. If you find that `/etc/resolv.conf` is not being updated, please check to make sure that the link is intact.

The GNU C Library resolver library isn't the only resolver library available. However, any resolver library that reads `/etc/resolv.conf` (and most of them do, in order to be compatible) should work fine with resolvconf.

Subscriber packages that need to know only when the resolver configuration file has changed should install a script in `/etc/resolvconf/update-libc.d/` rather than in `/etc/resolvconf/update.d/`. (For example, two packages that install *update-libc.d/* hook scripts are **fetchmail** and **squid**.) This is important for synchronization purposes: scripts in *update-libc.d/* are run *after* `resolv.conf` has been updated; the same is not necessarily true of scripts in *update.d/*.

OPTIONS

-a *IFACE.PROG*

Add or overwrite the record *IFACE.PROG* then run the update scripts if updating is enabled. When this option is used the information must be provided to

resolvconf on its standard input in the format of the [resolv.conf\(5\)](#) file. Each line in the file must be terminated by a newline.

-d *IFACE.PROG*

Delete the record *IFACE.PROG* then run the update scripts if updating is enabled.

The string *IFACE.PROG* may not contain spaces, slashes, an initial dot, an initial hyphen or an initial tilde. It is conventionally formed from *IFACE*, the name of the interface involved, a dot, and *IPROG*, the name of the interface configuration program, e.g., "eth0.dhclient".

-u

Just run the update scripts (if updating is enabled).

With **-a**, **-d** or **-u**:

if updating is not enabled, schedule a delayed update. The delayed update will be carried out when updates are enabled.

--enable-updates

Set the flag indicating that **resolvconf** should run update scripts when invoked in the future with **-a**, **-d** or **-u**. If a delayed update was scheduled then run update scripts.

--disable-updates

Clear the flag.

--updates-are-enabled

Return 0 if the flag is set, otherwise return 1.

ENVIRONMENT VARIABLES

The following variables can be set in the configuration file */etc/default/resolvconf*. If the file does not exist you will have to create it.

REPORT_ABSENT_SYMLINK

If set to "yes" then **resolvconf** will print a message when */etc/resolv.conf* is not a symbolic link to the **resolvconf**-generated resolver configuration file. Set to "no" to prevent the printing of this message. The default is "yes".

TRUNCATE_NAMESERVER_LIST_AFTER_LOOPBACK_ADDRESS

If set to "yes" then the *libc* script will include no more nameserver addresses after the first nameserver address that is a loopback address. (In IPv4 a loopback address is any one that starts with "127.". In IPv6 the loopback address is "::1".)

The advantage of truncating the nameserver list after a loopback address is that doing so inhibits unnecessary changes to *resolv.conf* and thus reduces the number of instances in which the *update-libc.d/* scripts have to be run. When an interface is brought up or down the local caching nameserver that listens on the loopback address is still informed of the change and adapts accordingly; the clients of the resolver which use the local caching nameserver do not need to be notified of the change. A disadvantage of this mode of operation is that applications have no secondary or tertiary nameserver address to fall back on should the local caching nameserver crash. Insofar as a local nameserver crash

can be regarded as an unlikely event, this is a relatively minor disadvantage. Set to "no" to disable truncation. The default is "yes".

A deprecated synonym for this variable is `TRUNCATE_NAMESERVER_LIST_AFTER_127`.

FILES

/etc/default/resolvconf

See the `ENVIRONMENT VARIABLES` section.

/etc/resolvconf/run

This is a symbolic link to a location where nameserver information is stored. The location must be on a filesystem that is writable early in the boot sequence. In Debian the default location is `/run/resolvconf` and in the future this will be the only supported location; configurability of the location via `/etc/resolvconf/run` will be dropped. Nevertheless, clients should not make any assumptions about the location or the canonical path of this directory or the hierarchy that is constructed under it.

/etc/resolvconf/interface-order

Determines the order of precedence of nameserver addresses and search domain names. See above and [interface-order\(5\)](#).

/etc/resolvconf/resolv.conf.d/base

File containing basic resolver information. The lines in this file are included in the resolver configuration file even when no interfaces are configured.

/etc/resolvconf/resolv.conf.d/head

File to be prepended to the dynamically generated resolver configuration file. Normally this is just a comment line.

/etc/resolvconf/resolv.conf.d/tail

File to be appended to the dynamically generated resolver configuration file. To append nothing, make this an empty file. This file is a good place to put a resolver **options** line if one is needed, e.g.,

```
options inet6
```

/etc/resolvconf/resolv.conf.d/original

Copy of the `/etc/resolv.conf` file before the `resolvconf` package was installed. This file has no effect on the functioning of `resolvconf`; it is retained so that `/etc/resolv.conf` can be restored to its original state if the `resolvconf` package is removed.

Note also that a copy of this file is included in the database until the first reboot after installation of the `resolvconf` package; this ensures that nameservers reachable before installation of `resolvconf` are still reachable after installation of `resolvconf` even though at that point not all suppliers of nameserver information may have supplied their information to [resolvconf\(8\)](#).

Note also that the administrator can choose to create a symbolic link in `/etc/resolvconf/resolv.conf.d/` from *tail* to *original* so that the contents of *original* are always added to the end of the dynamically generated file.

BUGS

Currently **resolvconf** does not check the sanity of the information provided to it.

AUTHOR

Written by Thomas Hood <jdthood@gmail.com> with contributions by Nathan Stratton Treadway.

COPYRIGHT

Copyright © 2004-2013 Thomas Hood.

This is free software; see the source for copying conditions.

SEE ALSO

[interface-order\(5\)](#), [resolv.conf\(5\)](#), [resolver\(3\)](#).

Read the resolvconf package *README* file for more in-depth information.

3 Feb 2013

resolvconf