NAME

resolvconf - a framework for managing multiple DNS configurations

SYNOPSIS

resolvconf -I
resolvconf [**-m** *metric*] [**-p**] [**-x**] **-a** *interface*[.*protocol*] <*file*
resolvconf [**-f**] **-d** *interface*[.*protocol*]
resolvconf [**-x**] **-il** *pattern*
resolvconf -u
resolvconf --version

DESCRIPTION

resolvconf manages resolv.conf(5) files from multiple sources, such as DHCP and VPN clients. Traditionally, the host runs just one client and that updates */etc/resolv.conf*. More modern systems frequently have wired and wireless interfaces and there is no guarantee both are on the same network. With the advent of VPN and other types of networking daemons, many things now contend for the contents of */etc/resolv.conf*.

resolvconf solves this by letting the daemon send their resolv.conf(5) file to **resolvconf** via stdin(4) with the argument **-a** *interface*[.*protocol*] instead of the filesystem. **resolvconf** then updates */etc/resolv.conf* as it thinks best. When a local resolver other than libc is installed, such as dnsmasq(8) or named(8), then **resolvconf** will supply files that the resolver should be configured to include.

resolvconf assumes it has a job to do. In some situations **resolvconf** needs to act as a deterrent to writing to */etc/resolv.conf*. Where this file cannot be made immutable or you just need to toggle this behaviour, **resolvconf** can be disabled by adding **resolvconf**=NO to resolvconf.conf(5).

resolvconf can mark an interfaces *resolv.conf* as private. This means that the name servers listed in that *resolv.conf* are only used for queries against the domain/search listed in the same file. This only works when a local resolver other than libc is installed. See resolvconf.conf(5) for how to configure **resolvconf** to use a local name server and how to remove the private marking.

resolvconf can mark an interfaces *resolv.conf* as exclusive. Only the latest exclusive interface is used for processing, otherwise all are.

When an interface goes down, it should then call **resolvconf** with **-d** *interface.** arguments to delete the *resolv.conf* file(s) for all the *protocols* on the *interface*.

Here are some options for the above commands:-

-f   Ignore non existent interfaces. Only really useful for deleting interfaces.

-m *metric*   Set the metric of the interface when adding it, default of 0. Lower metrics take precedence.
                 This affects the default order of interfaces when listed.

-p   Marks the interface *resolv.conf* as private.

-x   Mark the interface *resolv.conf* as exclusive when adding, otherwise only use the latest
                 exclusive interface.

resolvconf has some more commands for general usage:-

-i *pattern* List the interfaces and protocols, optionally matching *pattern*, we have *resolv.conf* files for.

-l *pattern* List the *resolv.conf* files we have. If *pattern* is specified then we list the files for the
                 interfaces and protocols that match it.

-u   Force **resolvconf** to update all its subscribers. **resolvconf** does not update the subscribers
                 when adding a resolv.conf that matches what it already has for that interface.

--version   Echo the resolvconf version to *stdout*.

resolvconf also has some commands designed to be used by it's subscribers and system startup:-

-I   Initialise the state directory */var/run/resolvconf*. This only needs to be called if the initial
                 system boot sequence does not automatically clean it out; for example the state directory is
                 moved somewhere other than */var/run*. If used, it should only be called once as early in the
                 system boot sequence as possible and before **resolvconf** is used to add interfaces.

-R   Echo the command used to restart a service.

-r *service* If the *service* is running then restart it. If the service does not exist or is not running then
                 zero is returned, otherwise the result of restarting the service.

-v   Echo variables DOMAINS, SEARCH and NAMESERVERS so that the subscriber can
                 configure the resolver easily.

-V   Same as **-v** except that only the information configured in resolvconf.conf(5) is set.

INTERFACE ORDERING

For **resolvconf** to work effectively, it has to process the resolv.confs for the interfaces in the correct order. **resolvconf** first processes interfaces from the **interface_order** list, then interfaces without a metic and that match the **dynamic_order** list, then interfaces with a metric in order and finally the rest in the operating systems lexical order. See resolvconf.conf(5) for details on these lists.

PROTOCOLS

Here are some suggested protocol tags to use for each *resolv.conf* file registered on an *interface*:-

dhcp   Dynamic Host Configuration Protocol. Initial versions of **resolvconf** did not recommend a *protocol* tag be appended to the *interface* name. When the protocol is absent, it is assumed to be the DHCP protocol.

ppp   Point-to-Point Protocol.

ra   IPv6 Router Advertisement.

dhcp6   Dynamic Host Configuration Protocol, version 6.

IMPLEMENTATION NOTES

If a subscriber has the executable bit then it is executed otherwise it is assumed to be a shell script and sourced into the current environment in a subshell. This is done so that subscribers can remain fast, but are also not limited to the shell language.

Portable subscribers should not use anything outside of */bin* and */sbin* because */usr* and others may not be available when booting. Also, it would be unwise to assume any shell specific features.

ENVIRONMENT

IF_METRIC
If the **-m** option is not present then we use *IF_METRIC* for the metric.

IF_PRIVATE
Marks the interface *resolv.conf* as private.

IF_EXCLUSIVE
Marks the interface *resolv.conf* as exclusive.

FILES

/etc/resolv.conf.bak
Backup file of the original resolv.conf.

/etc/resolvconf.conf
Configuration file for **resolvconf**.

/libexec/resolvconf
Directory of subscribers which are run every time **resolvconf** adds, deletes or updates.

/libexec/resolvconf/libc.d
Directory of subscribers which are run after the libc subscriber is run.

/var/run/resolvconf
State directory for **resolvconf**.

SEE ALSO

resolver(3), stdin(4), resolv.conf(5), resolvconf.conf(5)

HISTORY

This implementation of **resolvconf** is called openresolv and is fully command line compatible with Debian's resolvconf, as written by Thomas Hood.

AUTHORS

Roy Marples *<roy@marples.name>*

BUGS

Please report them to **http://roy.marples.name/projects/openresolv**

resolvconf does not validate any of the files given to it.

When running a local resolver other than libc, you will need to configure it to include files that **resolvconf** will generate. You should consult resolvconf.conf(5) for instructions on how to configure your resolver.