NAME

dhcpcd — a DHCP client

SYNOPSIS

dhcpcd [−146ABbDdEGgHJKLMNPpqTV] [−C, −−nohook *hook*] [−c, −−script *script*]
[−e, −−env *value*] [−F, −−fqdn *FQDN*] [−f, −−config *file*]
[−h, −−hostname *hostname*] [−I, −−clientid *clientid*]
[−i, −−vendorclassid *vendorclassid*] [−j, −−logfile *logfile*]
[−l, −−leasetime *seconds*] [−m, −−metric *metric*]
[−O, −−nooption *option*] [−o, −−option *option*] [−Q, −−require *option*]
[−r, −−request *address*] [−S, −−static *value*]
[−s, −−inform *address*[*/cidr*[*/broadcast_address*]]] [−−inform6]
[−t, −−timeout *seconds*] [−u, −−userclass *class*]
[−v, −−vendor *code*, *value*] [−W, −−whitelist *address*[*/cidr*]] [−w]
[−−waitip=[4 | 6]] [−y, −−reboot *seconds*]
[−X, −−blacklist *address*[*/cidr*]] [−Z, −−denyinterfaces *pattern*]
[−z, −−allowinterfaces *pattern*] [−−inactive] [−−configure]
[−−noconfigure] [interface] [...]
dhcpcd −n, −−rebind [interface]
dhcpcd −k, −−release [interface]
dhcpcd −U, −−dumplease [*interface*]
dhcpcd −−version
dhcpcd −x, −−exit [interface]

DESCRIPTION

dhcpcd is an implementation of the DHCP client specified in RFC 2131. **dhcpcd** gets the host information (IP address, routes, etc) from a DHCP server and configures the network *interface* of the machine on which it is running. **dhcpcd** then runs the configuration script which writes DNS information to *resolvconf*(8), if available, otherwise directly to */etc/resolv.conf*. If the hostname is currently blank, (null) or localhost, or *force_hostname* is YES or TRUE or 1 then **dhcpcd** sets the hostname to the one supplied by the DHCP server. **dhcpcd** then daemonises and waits for the lease renewal time to lapse. It will then attempt to renew its lease and reconfigure if the new lease changes when the lease begins to expire or the DHCP server sends a message to renew early.

If any interface reports a working carrier then **dhcpcd** will try to obtain a lease before forking to the background, otherwise it will fork right away. This behaviour can be modified with the −b, −−background and −w, −−waitip options.

dhcpcd is also an implementation of the BOOTP client specified in RFC 951.

dhcpcd is also an implementation of the IPv6 Router Solicitor as specified in RFC 4861 and RFC 6106.

dhcpcd is also an implementation of the IPv6 Privacy Extensions to AutoConf as specified in RFC 4941. This feature needs to be enabled in the kernel and **dhcpcd** will start using it.

dhcpcd is also an implementation of the DHCPv6 client as specified in RFC 3315. By default, **dhcpcd** only starts DHCPv6 when instructed to do so by an IPV6 Router Advertisement. If no Identity Association is configured, then a Non-temporary Address is requested.

Local Link configuration

If **dhcpcd** failed to obtain a lease, it probes for a valid IPv4LL address (aka ZeroConf, aka APIPA). Once obtained it restarts the process of looking for a DHCP server to get a proper address.

When using IPv4LL, **dhcpcd** nearly always succeeds and returns an exit code of 0. In the rare case it fails, it normally means that there is a reverse ARP proxy installed which always defeats IPv4LL probing. To disable this behaviour, you can use the −L, −−noipv4ll option.

Multiple interfaces

If a list of interfaces are given on the command line, then **dhcpcd** only works with those interfaces, otherwise **dhcpcd** discovers available Ethernet interfaces that can be configured. When **dhcpcd** is not limited to one interface on the command line, it is running in Manager mode. The **dhcpcd-ui** project expects dhcpcd to be running this way.

If a single interface is given then **dhcpcd** only works for that interface and runs as a separate instance to other **dhcpcd** processes. The `-w`, `--waitip` option is enabled in this instance to maintain compatibility with older versions. Using a single interface, optionally further limited to an address protocol, also affects the `-k`, `-N`, `-n` and `-x` options, where the same interface and any address protocol will need to be specified, as a lack of an interface will imply Manager mode which this is not. To force starting in Manager mode with only one interface, the `-M`, `--manager` option can be used.

Interfaces are preferred by carrier, DHCP lease/IPv4LL and then lowest metric. For systems that support route metrics, each route will be tagged with the metric, otherwise **dhcpcd** changes the routes to use the interface with the same route and the lowest metric. See options below for controlling which interfaces we allow and deny through the use of patterns.

Non-ethernet interfaces and some virtual ethernet interfaces such as TAP and bridge are ignored by default, as is the FireWire interface. To work with these devices they either need to be specified on the command line, be listed in `--allowinterfaces` or have an interface directive in */etc/dhcpcd.conf*.

Hooking into events

dhcpcd runs */lib/dhcpcd/dhcpcd-run-hooks*, or the script specified by the `-c`, `--script` option. This script runs each script found in */lib/dhcpcd/dhcpcd-hooks* in a lexical order. The default installation supplies the scripts *01-test*, *20-resolv.conf* and *30-hostname*. You can disable each script by using the `-C`, `--nohook` option. See *dhcpcd-run-hooks*(8) for details on how these scripts work. **dhcpcd** currently ignores the exit code of the script.

More scripts are supplied in */usr/share/dhcpcd/hooks* and need to be copied to */lib/dhcpcd/dhcpcd-hooks* if you intend to use them. For example, you could install *29-lookup-hostname* so that **dhcpcd** can lookup the hostname of the IP address in DNS if no hostname is given by the lease and one is not already set.

Fine tuning

You can fine-tune the behaviour of **dhcpcd** with the following options:

`-b`, `--background`
: Background immediately. This is useful for startup scripts which don't disable link messages for carrier status.

`-c`, `--script` *script*
: Use this *script* instead of the default */lib/dhcpcd/dhcpcd-run-hooks*.

`-D`, `--duid` [*ll* | *lt* | *uuid* | *value*]
: Use a DHCP Unique Identifier. If persistent storage is available then a DUID-LLT (link local address + time) is generated, otherwise DUID-LL is generated (link local address). The DUID type can be hinted as an optional parameter if the file */var/lib/dhcpcd/duid* does not exist. If not*ll*, *lt* or *uuid* then *value* will be converted from 00:11:22:33 format. This, plus the IAID will be used as the `-I`, `--clientid`. The DUID generated will be held in*/var/lib/dhcpcd/duid* and should not be copied to other hosts. This file also takes precedence over the above rules except for setting a value.

`-d`, `--debug`
: Echo debug messages to the stderr and syslog.

`-E`, `--lastlease`
: If **dhcpcd** cannot obtain a lease, then try to use the last lease acquired for the interface.

`--lastleaseextend`
: Same as the above, but the lease will be retained even if it expires. **dhcpcd** will give it up if any other host tries to claim it for their own via ARP. This violates RFC 2131, section 3.7, which

states the lease should be dropped once it has expired.

`-e`, `--env` *`value`*
: Push *`value`* to the environment for use in *dhcpcd-run-hooks*(8). For example, you can force the hostname hook to always set the hostname with `-e` *force_hostname=YES*.

`-g`, `--reconfigure`
: **dhcpcd** will re-apply IP address, routing and run *dhcpcd-run-hooks*(8) for each interface. This is useful so that a 3rd party such as PPP or VPN can change the routing table and / or DNS, etc and then instruct **dhcpcd** to put things back afterwards. **dhcpcd** does not read a new configuration when this happens - you should rebind if you need that functionality.

`-F`, `--fqdn` *`fqdn`*
: Requests that the DHCP server update DNS using FQDN instead of just a hostname. Valid values for *`fqdn`* are disable, none, ptr and both. **dhcpcd** itself never does any DNS updates. **dhcpcd** encodes the FQDN hostname as specified in `RFC 1035`.

`-f`, `--config` *`file`*
: Specify a config to load instead of */etc/dhcpcd.conf*. **dhcpcd** always processes the config file before any command line options.

`-h`, `--hostname` *`hostname`*
: Sends *`hostname`* to the DHCP server so it can be registered in DNS. If *`hostname`* is an empty string then the current system hostname is sent. If *`hostname`* is a FQDN (i.e., contains a .) then it will be encoded as such.

`-I`, `--clientid` *`clientid`*
: Send the *`clientid`*. If the string is of the format 01:02:03 then it is encoded as hex. For interfaces whose hardware address is longer than 8 bytes, or if the *`clientid`* is an empty string then **dhcpcd** sends a default *`clientid`* of the hardware family and the hardware address.

`-i`, `--vendorclassid` *`vendorclassid`*
: Override the DHCPv4 *`vendorclassid`* field sent. The default is dhcpcd-<version>:<os>:<machine>:<platform>. For example
  dhcpcd-5.5.6:NetBSD-6.99.5:i386:i386
  If not set then none is sent. Some badly configured DHCP servers reject unknown vendorclassids. To work around it, try and impersonate Windows by using the MSFT vendorclassid.

`-j`, `--logfile` *`logfile`*
: Writes to the specified *`logfile`*. **dhcpcd** still writes to *syslog*(3). The *`logfile`* is reopened when **dhcpcd** receives the `SIGUSR2` signal.

`-k`, `--release` [*`interface`*]
: This causes an existing **dhcpcd** process running on the *`interface`* to release its lease and deconfigure the *`interface`* regardless of the `-p`, `--persistent` option. If no *`interface`* is specified then this applies to all interfaces in Manager mode. If no interfaces are left running, **dhcpcd** will exit.

`-l`, `--leasetime` *`seconds`*
: Request a lease time of *`seconds`*. *`-1`* represents an infinite lease time. By default **dhcpcd** does not request any lease time and leaves it in the hands of the DHCP server.

`-M`, `--manager`
: Start **dhcpcd** in Manager mode even if only one interface specified on the command line. See the Multiple Interfaces section above.

`-m`, `--metric` *`metric`*
: Metrics are used to prefer an interface over another one, lowest wins. **dhcpcd** will supply a default metric of 1000 + *if_nametoindex*(3). This will be offset by 2000 for wireless interfaces, with additional offsets of 1000000 for IPv4LL and 2000000 for roaming interfaces.

`-n`, `--rebind` [*interface*]
Notifies **dhcpcd** to reload its configuration and rebind the specified *interface*. If no *interface* is specified then this applies to all interfaces in Manager mode. If **dhcpcd** is not running, then it starts up as normal.

`-N`, `--renew` [*interface*]
Notifies **dhcpcd** to renew existing addresses on the specified *interface*. If no*interface* is specified then this applies to all interfaces in Manager mode. If **dhcpcd** is not running, then it starts up as normal. Unlike the `-n`, `--rebind` option abo ve, the configuration for **dhcpcd** is not reloaded.

`-o`, `--option` *option*
Request the DHCP *option* variable for use in */lib/dhcpcd/dhcpcd-run-hooks*.

`-p`, `--persistent`
dhcpcd de-configures the *interface* when it exits unless this option is enabled. Sometimes, this isn't desirable if, for example, you have root mounted over NFS or SSH clients connect to this host and they need to be notified of the host shutting down. You can use this option to stop this from happening.

`-r`, `--request` *address*
Request the *address* in the DHCP DISCOVER message. There is no guarantee this is the address the DHCP server will actually give. If no*address* is gi ven then the first address currently assigned to the *interface* is used.

`-s`, `--inform` *address*[*/cidr*[*/broadcast_address*]]
Behaves like `-r`, `--request` as above, but sends a DHCP INFORM instead of DISCOVER/REQUEST. This does not get a lease as such, just notifies the DHCP server of the *address* in use. You should also include the optional *cidr* network number in case the address is not already configured on the interface. **dhcpcd** remains running and pretends it has an infinite lease. **dhcpcd** will not de-configure the interf ace when it exits. If**dhcpcd** f ails to contact a DHCP server then it returns a failure instead of falling back on IPv4LL.

`--inform6`
Performs a DHCPv6 Information Request. No address is requested or specified, but all other DHCPv6 options are allowed. This is normally performed automatically when the IPv6 Router Advertises that the client should perform this operation. This option is only needed when **dhcpcd** is not processing IPv6RA messages and the need for DHCPv6 Information Request exists.

`-S`, `--static` *value*
Configures a static DHCP *value*. If you set**ip_address** then **dhcpcd** will not attempt to obtain a lease and just use the value for the address with an infinite lease time.

Here is an example which configures a static address, routes and DNS.

```
dhcpcd -S ip_address=192.168.0.10/24 \
-S routers=192.168.0.1 \
-S domain_name_servers=192.168.0.1 \
eth0
```

You cannot presently set static DHCPv6 values. Use the`-e`, `--env` option instead.

`-t`, `--timeout` *seconds*
Timeout after *seconds*, instead of the default 30. On timeout, **dhcpcd** will exit if the `-1`, `--oneshot` option has been given, otherwise it will fork into the background and keep on trying. A setting of 0*seconds* causes **dhcpcd** to w ait forever to get a lease.

`-u`, `--userclass` *class*
Tags the DHCPv4 message with the userclass *class*. DHCP servers use this to give members of the class DHCP options other than the default, without having to know things like hardware

address or hostname.

`-v, --vendor` *code,value*
: Add an encapsulated vendor option. *code* should be between 1 and 254 inclusi ve. To add a raw vendor string, omit *code* but keep the comma. Examples.

  Set the vendor option 01 with an IP address.
  dhcpcd -v 01,192.168.0.2 eth0
  Set the vendor option 02 with a hex code.
  dhcpcd -v 02,01:02:03:04:05 eth0
  Set the vendor option 03 with an IP address as a string.
  dhcpcd -v 03,\"192.168.0.2\" eth0
  Set un-encapsulated vendor option to hello world.
  dhcpcd -v ,"hello world" eth0

`--version`
: Display both program version and copyright information. **dhcpcd** then e xits before doing any configuration.

`-w`
: Wait for an address to be assigned before forking to the background. Does not take an argument, unlike the below option.

`--waitip`=[4 | 6]
: Wait for an address to be assigned before forking to the background. 4 means wait for an IPv4 address to be assigned. 6 means wait for an IPv6 address to be assigned. If no argument is given, **dhcpcd** will wait for any address protocol to be assigned. It is possible to wait for more than one address protocol and **dhcpcd** will only fork to the background when all waiting conditions are satisfied.

`-x, --exit` [*interface*]
: This will signal an existing **dhcpcd** process running on the *interface* to exit. If no *interface* is specified, then the above is applied to all interfaces in Manager mode. See the `-p, --persistent` option to control configuration persistence on exit, which is enabled by default in *dhcpcd.conf*(5). **dhcpcd** then waits until this process has exited.

`-y, --reboot` *seconds*
: Allow *reboot* seconds before moving to the discover phase if we have an old lease to use. Allow *reboot* seconds before starting fallback states from the discover phase. IPv4LL is started when the first *reboot* timeout is reached. The default is 5 seconds. A setting of 0 seconds causes **dhcpcd** to skip the reboot phase and go straight into discover. This has no effect on DHCPv6 other than skipping the reboot phase.

Restricting behaviour

dhcpcd will try to do as much as it can by default. However, there are sometimes situations where you don't want the things to be configured exactly how the DHCP server wants. Here are some options that deal with turning these bits off.

Note that when **dhcpcd** is restricted to a single interface then the interface also needs to be specified when asking **dhcpcd** to exit using the commandline. If the protocol is restricted as well then the protocol needs to be included with the exit instruction.

`-1, --oneshot`
: Exit after configuring an interface. Use the `-w, --waitip` option to specify which protocol(s) to configure before exiting.

`-4, --ipv4only`
: Configure IPv4 only.

`-6`, `--ipv6only`
Configure IPv6 only.

`-A`, `--noarp`
Don't request or claim the address by ARP. This also disables IPv4LL.

`-B`, `--nobackground`
Don't run in the background when we acquire a lease. This is mainly useful for running under the control of another process, such as a debugger or a network manager.

`-C`, `--nohook` *`script`*
Don't run this hook script. Matches full name, or prefixed with 2 numbers optionally ending with *.sh*.

So to stop **dhcpcd** from touching your DNS settings you would do:-
dhcpcd -C resolv.conf eth0

`-G`, `--nogateway`
Don't set any default routes.

`-H`, `--xidhwaddr`
Use the last four bytes of the hardware address as the DHCP xid instead of a randomly generated number.

`-J`, `--broadcast`
Instructs the DHCP server to broadcast replies back to the client. Normally this is only set for non-Ethernet interfaces, such as FireWire and InfiniBand. In most instances, **dhcpcd** will set this automatically.

`-K`, `--nolink`
Don't receive link messages for carrier status. You should only have to use this with buggy device drivers or running **dhcpcd** through a network manager.

`-L`, `--noipv4ll`
Don't use IPv4LL (aka APIPA, aka Bonjour, aka ZeroConf).

`-O`, `--nooption` *`option`*
Removes the *`option`* from the DHCP message before processing.

`-P`, `--printpidfile`
Print the *pidfile* **dhcpcd** will use based on command-line arguments to stdout.

`-Q`, `--require` *`option`*
Requires the *`option`* to be present in all DHCP messages, otherwise the message is ignored. To enforce that **dhcpcd** only responds to DHCP servers and not BOOTP servers, you can `-Q` *`dhcp_message_type`*.

`-q`, `--quiet`
Quiet **dhcpcd** on the command line, only warnings and errors will be displayed. If this option is used another time then all console output is disabled. These messages are still logged via *syslog*(3).

`-T`, `--test`
On receipt of DHCP messages just call */lib/dhcpcd/dhcpcd-run-hooks* with the reason of TEST which echos the DHCP variables found in the message to the console. The interface configuration isn't touched and neither are any configuration files. The *`rapid_commit`* option is not sent in TEST mode so that the server does not lease an address. To test INFORM the interface needs to be configured with the desired address before starting **dhcpcd**.

`-U`, `--dumplease` [*`interface`*]
Dumps the current lease for the *`interface`* to stdout. If no *`interface`* is given then all interfaces are dumped. Use the `-4` or `-6` flags to specify an address family. If a lease is piped in

via standard input then use the special interface named `-` to dump it. In this case, specifying an address family is mandatory.

`-V`, `--variables`
: Display a list of option codes, the associated variable and encoding for use in *dhcpcd-run-hooks*(8). Variables are prefixed with new_ and old_ unless the option number is -. Variables without an option are part of the DHCP message and cannot be directly requested.

`-W`, `--whitelist` *address*[/cidr]
: Only accept packets from *address*[/cidr]. `-X`, `--blacklist` is ignored if `-W`, `--whitelist` is set.

`-X`, `--blacklist` *address*[*/cidr*]
: Ignore all packets from *address*[*/cidr*].

`-Z`, `--denyinterfaces` *pattern*
: When discovering interfaces, the interface name must not match *pattern* which is a space or comma separated list of patterns passed to *fnmatch*(3).

`-z`, `--allowinterfaces` *pattern*
: When discovering interfaces, the interface name must match *pattern* which is a space or comma separated list of patterns passed to *fnmatch*(3). If the same interface is matched in `-Z`, `--denyinterfaces` then it is still denied.

`--inactive`
: Don't start any interfaces other than those specified on the command line. This allows **dhcpcd** to be started in Manager mode and then wait for subsequent **dhcpcd** commands to start each interface as required.

`--configure`
: Allows **dhcpcd** to configure the system. This is the default behaviour and sets `if_configured=true`.

`--noconfigure`
: **dhcpcd** will not configure the system at all. This is only of use if the `--script` that **dhcpcd** calls at each network event configures the system instead. This is different from `-T`, `--test` mode in that it's not one shot and the only change to the environment is the addition of `if_configured=false`.

`--nodev`
: Don't load any */dev* management modules.

3RDPARTY LINK MANAGEMENT

Some interfaces require configuration by 3rd parties, such as PPP or VPN. When an interface configuration in **dhcpcd** is marked as STATIC or INFORM without an address then **dhcpcd** will monitor the interface until an address is added or removed from it and act accordingly. For point to point interfaces (like PPP), a default route to its destination is automatically added to the configuration. If the point to point interface is configured for INFORM, then **dhcpcd** unicasts INFORM to the destination, otherwise it defaults to STATIC.

NOTES

dhcpcd requires a Berkeley Packet Filter, or BPF device on BSD based systems and a Linux Socket Filter, or LPF device on Linux based systems for all IPv4 configuration.

If restricting **dhcpcd** to a single interface and optionally address family via the command-line then all further calls to **dhcpcd** to rebind, reconfigure or exit need to include the same restrictive flags so that **dhcpcd** knows which process to signal.

Some DHCP servers implement ClientID filtering. If **dhcpcd** is replacing an in-use DHCP client then you might need to adjust the clientid option **dhcpcd** sends to match. If using a DUID in place of the ClientID, edit */var/lib/dhcpcd/duid* accordingly.

FILES

/etc/dhcpcd.conf
Configuration file for dhcpcd. If you always use the same options, put them here.

/lib/dhcpcd/dhcpcd-run-hooks
Bourne shell script that is run to configure or de-configure an interface.

/lib64/dhcpcd/dev
Linux */dev* management modules.

/lib/dhcpcd/dhcpcd-hooks
A directory containing Bourne shell scripts that are run by the above script. Each script can be disabled by using the `-C`, `--nohook` option described above.

/var/lib/dhcpcd/duid
Text file that holds the DUID used to identify the host.

/var/lib/dhcpcd/secret
Text file that holds a secret key known only to the host.

/var/lib/dhcpcd/`interface-ssid`.lease
The actual DHCP message sent by the server. We use this when reading the last lease and use the file's mtime as when it was issued.

/var/lib/dhcpcd/`interface-ssid`.lease6
The actual DHCPv6 message sent by the server. We use this when reading the last lease and use the file's mtime as when it was issued.

/var/lib/dhcpcd/rdm_monotonic
Stores the monotonic counter used in the *replay* field in Authentication Options.

/run/dhcpcd/pid
Stores the PID of **dhcpcd** running on all interfaces.

/run/dhcpcd/`interface`.pid
Stores the PID of **dhcpcd** running on the *interface*.

/run/dhcpcd/sock
Control socket to the manager daemon.

/run/dhcpcd/unpriv.sock
Unprivileged socket to the manager daemon, only allows state retrieval.

/run/dhcpcd/`interface`.sock
Control socket to per interface daemon.

/run/dhcpcd/`interface`.unpriv.sock
Unprivileged socket to per interface daemon, only allows state retrieval.

SEE ALSO

fnmatch(3), *if_nametoindex*(3), *dhcpcd.conf*(5), *resolv.conf*(5), *dhcpcd-run-hooks*(8), *resolvconf*(8)

STANDARDS

RFC 951, RFC 1534, RFC 2104, RFC 2131, RFC 2132, RFC 2563, RFC 2855, RFC 3004, RFC 3118, RFC 3203, RFC 3315, RFC 3361, RFC 3633, RFC 3396, RFC 3397, RFC 3442, RFC 3495, RFC 3925, RFC 3927, RFC 4039, RFC 4075, RFC 4242, RFC 4361, RFC 4390, RFC 4702, RFC 4074, RFC 4861, RFC 4833, RFC 4941, RFC 5227, RFC 5942, RFC 5969, RFC 6106, RFC 6334, RFC 6355, RFC 6603, RFC 6704, RFC 7217, RFC 7550, RFC 7844.

AUTHORS

Roy Marples <*roy@marples.name*>

BUGS

Please report them to https://roy.marples.name/projects/dhcpcd