

NAME

dhcpcd-run-hooks — DHCP client configuration script

DESCRIPTION

dhcpcd-run-hooks is used by *dhcpcd*(8) to run any system and user defined hook scripts. System hook scripts are found in */lib/dhcpcd/dhcpcd-hooks* and the user defined hooks are */etc/dhcpcd.enter-hook*, and */etc/dhcpcd.exit-hook*. The default install supplies hook scripts for configuring */etc/resolv.conf* and the hostname. Your distribution may have included other hook scripts to say configure ntp or ypbind. A test hook is also supplied that simply echos the dhcp variables to the console from DISCOVER message.

The hooks scripts are loaded into the current shell rather than executed in their own process. This allows each hook script, such as */etc/dhcpcd.enter-hook* to customise environment variables or provide alternative functions to hooks further down the chain. As such, using the shell builtins **exit**, **exec** or similar will cause **dhcpcd-run-hooks** to exit at that point.

Each time **dhcpcd-run-hooks** is invoked, `$interface` is set to the interface that **dhcpcd** is run on and `$reason` is set to the reason why **dhcpcd-run-hooks** was invoked. DHCP information to be configured is held in variables starting with the word `new_` and old DHCP information to be removed is held in variables starting with the word `old_`. **dhcpcd** can display the full list of variables it knows about by using the `-V`, `--variables` argument.

Here's a list of reasons why **dhcpcd-run-hooks** could be invoked:

PREINIT	dhcpcd is starting up and any pre-initialisation required should be performed now.
CARRIER	dhcpcd has detected the carrier is up. This is generally just a notification and no action need be taken.
NOCARRIER	dhcpcd lost the carrier. The cable may have been unplugged or association to the wireless point lost.
NOCARRIER_ROAMING	dhcpcd lost the carrier but the interface configuration is persisted. The OS has to support wireless roaming or IP Persistence for this to happen.
INFORM INFORM6	dhcpcd informed a DHCP server about its address and obtained other configuration details.
BOUND BOUND6	dhcpcd obtained a new lease from a DHCP server.
RENEW RENEW6	dhcpcd renewed its lease.
REBIND REBIND6	dhcpcd has rebound to a new DHCP server.
REBOOT REBOOT6	dhcpcd successfully requested a lease from a DHCP server.
RELEASE RELEASE6	dhcpcd has released the lease.
DELEGATED6	dhcpcd assigned a delegated prefix to the interface.
IPV4LL	dhcpcd obtained an IPV4LL address, or one was removed.
STATIC	dhcpcd has been configured with a static configuration which has not been obtained from a DHCP server.
3RDPARTY	dhcpcd is monitoring the interface for a 3rd party to give it an IP address.
TIMEOUT TIMEOUT6	dhcpcd failed to contact any DHCP servers but was able to use an old lease.
EXPIRE EXPIRE6	dhcpcd's lease or state expired and it failed to obtain a new one.

NAK	dhcpcd received a NAK from the DHCP server. This should be treated as EXPIRE.
RECONFIGURE	dhcpcd has been instructed to reconfigure an interface.
ROUTERADVERT	dhcpcd has received an IPv6 Router Advertisement, or one has expired.
STOP STOP6	dhcpcd stopped running on the interface.
STOPPED	dhcpcd has stopped entirely.
DEPARTED	The interface has been removed.
FAIL	dhcpcd failed to operate on the interface. This normally happens when dhcpcd does not support the raw interface, which means it cannot work as a DHCP or ZeroConf client. Static configuration and DHCP INFORM is still allowed.
TEST	dhcpcd received an OFFER from a DHCP server but will not configure the interface. This is primarily used to test the variables are filled correctly for the script to process them.

ENVIRONMENT

dhcpcd will clear the environment variables aside from `$PATH`. The following variables will then be set, along with any protocol supplied ones.

<code>\$interface</code>	the name of the interface.
<code>\$protocol</code>	the protocol that triggered the event.
<code>\$reason</code>	as described above.
<code>\$pid</code>	the pid of dhcpcd .
<code>\$ifcarrier</code>	the link status of <code>\$interface</code> : unknown, up or down.
<code>\$ifmetric</code>	<code>\$interface</code> preference, lower is better.
<code>\$ifwireless</code>	1 if <code>\$interface</code> is wireless, otherwise 0.
<code>\$ifflags</code>	<code>\$interface</code> flags.
<code>\$ifmtu</code>	<code>\$interface</code> MTU.
<code>\$ifssid</code>	the SSID the <code>interface</code> is connected to.
<code>\$interface_order</code>	A list of interfaces, in order of preference.
<code>\$if_configured</code>	true if dhcpcd has configured the interface otherwise false.
<code>\$if_up</code>	true if the interface is up, otherwise false. This is more than IFF_UP and may not be equal.
<code>\$if_down</code>	true if the interface is down, otherwise false. This is more than IFF_UP and may not be equal.
<code>\$af_waiting</code>	Address family waiting for, as defined in <code>dhcpcd.conf(5)</code> .
<code>\$profile</code>	the name of the profile selected from <code>dhcpcd.conf(5)</code> .
<code>\$new_delegated_dhcp6_prefix</code>	space-separated list of delegated prefixes.

FILES

When **dhcpcd-run-hooks** runs, it loads `/etc/dhcpcd.enter-hook`, any scripts found in `/lib/dhcpcd/dhcpcd-hooks` in lexical order, then finally `/etc/dhcpcd.exit-hook`.

SEE ALSO

dhcpcd(8)

AUTHORS

Roy Marples <*roy@marples.name*>

BUGS

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

SECURITY CONSIDERATIONS

dhcpcd will validate the content of each option against its encoding. For string, ascii, raw or binhex encoding it's up to the user to validate it for the intended purpose.

When used in a shell script, each variable must be quoted correctly.