NAME

dhclient-script - DHCP client network configuration script

DESCRIPTION

The DHCP client network configuration script is invoked from time to time by dhclient(8). This script is used by the DHCP client to set each interface's initial configuration prior to requesting an address, to test the address once it has been offered, and to set the interface's final configuration once a lease has been acquired. If no lease is acquired, the script is used to test predefined leases, if any, and also called once if no valid lease can be identified.

In general, customizations specific to a particular computer should be done in the */etc/dhclient.conf* file.

OPERATION

When dhclient(8) needs to invoke the client configuration script, it sets up a number of environment variables and runs **dhclient-script**. In all cases, *$reason* is set to the name of the reason why the script has been invoked. The following reasons are currently defined: MEDIUM, PREINIT, ARPCHECK, ARPSEND, BOUND, RENEW, REBIND, REBOOT, EXPIRE, FAIL and TIMEOUT.

MEDIUM The DHCP client is requesting that an interface's media type be set. The interface name is passed in *$interface*, and the media type is passed in *$medium*.

PREINIT The DHCP client is requesting that an interface be configured as required in order to send packets prior to receiving an actual address. This means configuring the interface with an IP address of 0.0.0.0 and a broadcast address of 255.255.255.255. The interface name is passed in *$interface*, and the media type in *$medium*.

If an IP alias has been declared in dhclient.conf(5), its address will be passed in *$alias_ip_address*, and that IP alias should be deleted from the interface, along with any routes to it.

ARPSEND The DHCP client is requesting that an address that has been offered to it be checked to see if somebody else is using it, by sending an ARP request for that address. It is not clear how to implement this, so no examples exist yet. The IP address to check is passed in *$new_ip_address*, and the interface name is passed in *$interface*.

ARPCHECK The DHCP client wants to know if a response to the ARP request sent using ARPSEND has been received. If one has, the script should exit with a nonzero status, indicating that the offered address has already been requested and should be declined. The *$new_ip_address* and *$interface* variables are set as with ARPSEND.

BOUND The DHCP client has done an initial binding to a new address. The new IP address is passed in *$new_ip_address*, and the interface name is passed in *$interface*. The media type is passed in *$medium*. Any options acquired from the server are passed using the option name described in dhcp-options(5), except that dashes ('-') are replaced by underscores ('_') in order to make valid shell variables, and the variable names start with "new_". So for example, the new subnet mask would be passed in *$new_subnet_mask*.

When a binding has been completed, a lot of network parameters are likely to need to be set up. A new */etc/resolv.conf* needs to be created, using the values of *$new_domain_name* and *$new_domain_name_servers* (which may list more than one server, separated by spaces). A default route should be set using *$new_routers*, and static routes may need to be set up using *$new_static_routes*.

If an IP alias has been declared, it must be set up here. The alias IP address will be written as *$alias_ip_address*, and other DHCP options that are set for the alias (e.g., subnet mask) will be passed in variables named as described previously except starting with "$alias_" instead of "$new_". Care should be taken that the alias IP address not be used if it is identical to the bound IP address (*$new_ip_address*), since the other alias parameters may be incorrect in this case.

RENEW When a binding has been renewed, the script is called as in BOUND, except that in addition to all the variables starting with "$new_", there is another set of variables starting with "$old_". Persistent settings that may have changed need to be deleted - for example, if a local route to the bound address is being configured, the old local route should be deleted. If the default route has changed, the old default route should be deleted. If the static routes have changed, the old ones should be deleted. Otherwise, processing can be done as with BOUND.

REBIND The DHCP client has rebound to a new DHCP server. This can be handled as with RENEW, except that if the IP address has changed, the ARP table should be cleared.

REBOOT The DHCP client has successfully reacquired its old address after a reboot. This can be processed as with BOUND.

EXPIRE The DHCP client has failed to renew its lease or acquire a new one, and the lease has expired. The IP address must be relinquished, and all related parameters should be deleted, as in RENEW and REBIND.

FAIL The DHCP client has been unable to contact any DHCP servers, and any leases that have been tested have not proved to be valid. The parameters from the last lease tested should

be deconfigured. This can be handled in the same way as EXPIRE.

TIMEOUT The DHCP client has been unable to contact any DHCP servers. However, an old lease has been identified, and its parameters have been passed in as with BOUND. The client configuration script should test these parameters and, if it has reason to believe they are valid, should exit with a value of zero. If not, it should exit with a nonzero value.

Before taking action according to *$reason*, **dhclient-script** will check for the existence of */etc/dhclient-enter-hooks*. If found, it will be sourced (see sh(1)). After taking action according to *$reason*, **dhclient-script** will check for the existence of */etc/dhclient-exit-hooks*. If found, it will be sourced (see sh(1)). These hooks scripts can be used to dynamically modify the environment at appropriate times during the DHCP negotiations. For example, if the administrator wishes to disable alias IP numbers on the DHCP interface, they might want to put the following in */etc/dhclient-enter-hooks*:

```
[ ."$reason" = .PREINIT ] && ifconfig $interface 0.0.0.0
```

The usual way to test a lease is to set up the network as with REBIND (since this may be called to test more than one lease) and then ping the first router defined in *$routers*. If a response is received, the lease must be valid for the network to which the interface is currently connected. It would be more complete to try to ping all of the routers listed in *$new_routers*, as well as those listed in *$new_static_routes*, but current scripts do not do this.

SEE ALSO

sh(1), dhclient.conf(5), dhclient.leases(5), dhclient(8), dhcpd(8), dhcrelay(8)

AUTHORS

The original version of **dhclient-script** was written for the Internet Software Consortium by Ted Lemon <*mellon@fugue.com*> in cooperation with Vixie Enterprises.

The OpenBSD implementation of **dhclient-script** was written by Kenneth R. Westerback <*krw@openbsd.org*>.

BUGS

If more than one interface is being used, there is no obvious way to avoid clashes between server-supplied configuration parameters - for example, the stock **dhclient-script** rewrites */etc/resolv.conf*. If more than one interface is being configured, */etc/resolv.conf* will be repeatedly initialized to the values provided by one server, and then the other. Assuming the information provided by both servers is valid, this should not cause any real problems, but it could be confusing.