

TESTING DHCP OPTION-60/43 – EXAMPLE SETUP

Application Note

v1.1

INTRODUCTION

The goal of this document includes the following:

- Describe how to enable DHCP “Vendor Class” options 60 and 43.
- Provide samples of configuration files correctly set for DHCP options 60 and 43
- Provide example vendorinfo executable script for handling DHCP option 43

OVERVIEW

On the WB platform, the default DHCP client used is BusyBox udhcp. This is managed by ifrc (our network interface-run-config utility) and the /etc/network/interfaces file.

The DHCP lease info is stored in the same format as that of isc-dhclient: /var/lib/dhcp/dhclient.<iface>.leases

Server response data is available via the client's local environment.

The DHCP events for udhcp are handled by /etc/dhcp/udhcp.script and further extensions may be added as source script hooks.

The client udhcp may be configured with the file: /etc/dhcp/udhcp.conf ...which contains settings for the various options and source script hooks.

In this example, the client will be sending a “vendorid”, using option 60, and also requesting an option 43 response. The server, if configured properly, will respond with formatted “vendorinfo” in option 43.

CONFIGURE DHCP OPTIONS 60 AND 43

DHCP “Vendor Class” options 60 and 43 are supported with the following settings:

In the file: **/etc/dhcp/udhcp.conf**

```
# vendor-class-identifier (opt60)
# if enabled, opt43 is added to the option request list
# set a string value here or specify a file (only the last line is read)
OPT_VCI=/root/vendorid
# source hook for vendorinfo-handler (opt43) script
VENDORINFO_HOOK=/root/vendorinfo
```

Note: The opt60 value is a variable length string of characters as specified by the client. This can be defined as a line of text either in the config file or inside another external file, used by the config file. If using an external file, the last line is parsed for the value and using the form name="value" is optional.

The value will appear in the udhcp process command line as: --vendorclass <text>

TESTING DHCP OPTION-60/43 – EXAMPLE SETUP – WB45NBT

Application Note

Example showing value set in the udhcpc.conf file:

```
OPT_VCI=Laird_WB45NBT
```

Example showing value set in an external file:

```
OPT_VCI=/root/vendorid
```

Note: When opt60 is set, then a request for option 43 is also set by the ifrc/udhcpc launcher. The response from the server is contained in local variable opt43 and may be processed by a source script hook. The response format is [`<length><data>`][...].

Example vendorid File for Setting Value of Option60

```
# /root/vendorid
# the vendor class identifier value may be quoted if necessary
#
Laird WB45NBT
```

Example Configuration on the isc-dhcpd Server

```
Add in file on server: /etc/dhcp/dhcpd.conf
option space WB;
option WB.server-address code 2 = ip-address;
option WB.server-name code 3 = text;
option WB.root-path code 4 = text;

class "VCI" {
    match option vendor-class-identifier;
}

# the option space values will only be used by matching vci
option WB.server-address 192.168.100.5;
option WB.server-name "example-isc-dhcpd";

subclass "VCI" "Laird WB45NBT" {
    vendor-option-space WB;
    option WB.root-path "/export/this/is/example/path-foo";
}
```

Example vendorinfo Executable Script for Handling Option43

```
# the following few lines before the shebang are optional and include
# some fake test data to allow testing this script if run standalone
fake=010868656c6c6f2e2e2e0204c0a89d6f0305667775637304192f6578706f72742f7468697
32f69732f6666f6f
data=${opt43:-$fake}

[ -n "$opt43" ] || echo opt43 not set, using fakedata...

#!/bin/ash
# /root/vendorinfo
# Parse option-43 response from DHCP server, when using option-60.
# jon.hefling@lairdtech.com

opt43len=${#data}
echo -e "opt43: $opt43len-bytes in message\n$data\n"
let p=0
while :
do
    id=${data:$p:2}
    len=${data:$p+2:2}
    [ -n "$id" -a -n "$len" ] || break

    # remove leading zeros and extract datum at p+2+2
    let len=${len/0/}
    text=${data:$p+4:$len*2}

    # display info
    echo -e "$id: $len-bytes\n$text"
    printf "\`echo -n $text |sed 's/\([0-9A-Fa-f]\{2\}\)/\\x\1/g'\`n"

    # skip ahead to next datum
    let p+=4
    let p+=len*2
done
echo
```

REVISION HISTORY

| Revision | Date | Description | Approved By |
|----------|--------------|------------------|-----------------|
| 1.0 | 2 July 2013 | Initial Version | John Imboden |
| 1.1 | 2 March 2014 | Formatting Edits | David Drogowski |
| | | | |