

## **Comdasy** FAQ: DHCP

Comdasy AG  
Rüdesheimer Str. 7  
D-80686 München  
Tel.: +49.89.5484333-0  
Fax: +49.89.5484333-29  
support@comdasy.com  
<http://www.comdasy.com>

## **Disclaimer**

We have taken all possible care to ensure that this manual contains correct, accurate information. However, the manufacturer cannot assume liability for any possible errors. In addition, the manufacturer cannot guarantee that the hardware will meet the purpose you require.

Comdasys reserves the right to make changes according to technical progress at any time. Brand names may be registered trademarks and must be treated as such.

© Copyright 2005-2008  
Comdasys AG  
80686 München, Germany

All rights reserved. No part of this manual may be reproduced, processed or distributed in any form (print, photocopy, microfilm or any other process) or processed by an electronic system without prior written permission from the manufacturer.

# Inhaltsverzeichnis

1. DHCP Options.....	4
2. Using DHCP client hooks.....	4
2.1. Introduction.....	4
2.2. Preparation.....	5
2.3. How the hook is called.....	5
2.4. Example hook.....	5
2.5. Additional notes.....	6
3. Troubleshooting.....	6

## 1. DHCP Options

The WebGUI only supports the configuration of standard options for the DHCP server. In many configurations, it is however necessary to supply custom options, or to even supply these only to certain devices. The configuration of such options can be done in the `/etc/sysconfig/dhcpd.conf` custom file. To define a hexadecimal option for example, use the following:

```
option someoption code 120 = string;
option someoption 00:08:66:69: ...;
```

In the same way, it is of course possible to define integer options (uint8,uint16,uint32,...) or IP addresses (ip-address).

After the definition of such an option, we can define rules as to who we want that option assigned to. In order to accomplish this, we define a class of devices identified in this case by the vendor ID.

```
class "foo" {
match if substring(option vendor-class-identifier, 0, 9) =
    "mediatrix";
option domain-name "bar.baz";
...
}
```

Then make a subnet declaration where some class is explicitly allowed. Explicitly denying a class is of course also possible. To do that, just use the keyword "deny" instead of "allow".

```
subnet 192.168.1.0 netmask 255.255.255.0 {
    allow members of "foo";
    range 192.168.1.10 192.168.1.200;
    option routers 192.168.1.1;
    option domain-name-servers 192.168.1.1;
}
```

If a subnet is to be subdivided into several parts where different options are to be assigned, pools can be used. Just define a pool inside the the subnet as seen in the following example.

```
pool {
    range 192.168.1.50 192.168.1.99;
    allow members of "foo";
    deny members of "bar";
}
```

## 2. Using DHCP client hooks

### 2.1. Introduction

The Comdasys Convergences use the ISC DHCP suite and its dhclient DHCP client. When this client processes a DHCP answer it can call a hook script which then is able to e.g. set values in

configuration files according to values sent by the DHCP server. This howto shows a simple example.

## 2.2. Preparation

Suppose we have a configuration file where we want to fill in our IP address and netmask that we've got from the DHCP server. Let's call this configuration file `/etc/foo.conf` and assume that it looks like this:

```
somedir=/foo/bar
text=This is just an example
ip=192.168.0.1
netmask=255.255.255.0
morestuff=no
```

The first step is to convert this file into a template where we can easily fill in values via regular expressions. In our example we're using DOS/Windows style pseudo-variables as they're so trivial to exchange. We'll call our new file `/etc/foo.conf.template`:

```
somedir=/foo/bar
text=This is just an example
ip=%IP%
netmask=%NETMASK%
morestuff=no
```

## 2.3. How the hook is called

Next we'll write the script that will transform the template into the configuration file. The ISC DHCP client looks for a shell script `/etc/dhclient-exit-hooks` and calls it when the DHCP client has done things like setting the machine's IP address and routing entries (see the *Additional notes* below for some details).

The DHCP client passes all values sent by the DHCP server via environment variables. You may want to write a test hook which just looks like the following example to see what the DHCP server sends and how the client names the variables:

```
env >/tmp/dhclient.env
```

Some variable names follow the simple scheme `new_dhcpoptionname` for the value sent by the server. For example, the names for the variables containing the IP address and netmask sent from the DHCP server for our machine to use are `new_ip_address` and `new_subnet_mask`.

## 2.4. Example hook

Back to our example configuration file and its template: to generate the configuration file `/etc/foo.conf` we'll write a `/etc/dhclient-exit-hooks` script that looks like this:

```
CONFIGFILE=/etc/foo.conf
CONFIGTEMPLATE=/etc/foo.conf.template

sed -e "s/%IP%/$new_ip_address/" \
    -e "s/%NETMASK%/$new_subnet_mask/" \
    <$CONFIGTEMPLATE >$CONFIGFILE
```

Depending on the application you may then also add commands to restart the application or

make it re-read its configuration file.

## 2.5. Additional notes

Some more in-depth informations about `dhclient-exit-hooks`: the script is actually not executed but "sourced", which means the parent script that calls the hook script reads the content of `dhclient-exit-hooks` and fills it in at a certain position of the parent script. This is why no "she-bang" (`#!/bin/sh` or `#!/bin/bash`) is needed and the hook script doesn't need to be executable.

Since the Comdasys Convergences only use bash the hook script must be in plain Bourne or bash style, C shell or Korn shell style hook scripts will fail !

## 3. Troubleshooting

Some updates require updating certain configuration files, as for example with the DHCP server component. Here, some updates clean up the leases file which needs to be reinitilaized for the DHCP server to start up correctly. The leases file can be found in `/var/state/dhcp/dhcpd.leases` . If the directory is missing, it just has to be recreated via:

```
mkdir /var/state/dhcp
```

Now the dhcp leases file can just be reinitialized via:

```
touch /var/state/dhcp/dhcpd.leases
```

The DHCP Server should now cleanly come up again.