I/A 50 Series and X-Terminals with Dynamic IP Addresses

Using DNS and DHCP

R. de Groot (Rob.de.Groot@IPS.Invensys.com)
R. Deen (Ron.Deen@IPS.Invensys.com)

Revision 1.5
March 15, 2005

Contents

Illustration Index

1 Introduction.

This manual attempts to explain how to setup **I/A 50 Series** hosts in combination with **MS-Windows** based X-Terminals having dynamic IP addresses for the Windows platform. **Please note that this is NOT an Official Foxboro manual. For support you are suggested to contact one of the authors. They may be able to help.** Configuring your X-Terminals is not a part of this manual. I suggest you have a look at the **X-Terminal Installation for Remote Display Manager and FoxView™** manual if that is what you need to do. This manual is a first attempt to setup and configure the **DNS** and **DHCP** services to work in this specific environment. This **DNS** and **DHCP** functionality is established on the "**Corporate Network**" side to which the **I/A 50 Series** system is connected through the secondary Ethernet™ interface only. We do appreciate any feedback regarding this subject. Sent your comments and remarks to the e-mail addresses provided on the cover page. Thanks in advance.

1.1 Disclaimer.

Due to the inherently complex nature of computer software, Invensys does not warrant that the software described in this document or this documentation is completely error free, will operate without interruption, is compatible with all equipment and software configurations, or will otherwise meet your needs. Accordingly, this documentation is provided as-is, and you assume all risks associated with its use. Invensys makes no warranties expressed or implied, regarding this document. In no event will Invensys be liable for indirect, incidental or consequential damages, including, without limitation, loss of income, use, or information.

1.2 To Do's

This is a work in progress but we can't think of anything right now.

1.3 Conventions.

The conventions used in this document are listed below.

When you see this:	It means this:
[Filename]	This typeface indicates a filename of which the contents are printed starting on the next line. This line is NOT part of the file contents.
`STATION#` **`some command<cr>`**	Text printed like this is ASCII text as it appears in a file or on screen. User data that has to be entered is printed in **bold**. Also used to display a pathname or filename in normal text. When used to indicate a command, type the bold printed command until <CR> which means to hit the ENTER key.
Use this data	This typeface is used to show a list of data to be entered in a location indicated in the text. Also used to describe a procedure.

When you see this:	It means this:
`<Alt_F4>`	Text displayed like this means: press the keys mentioned between brackets simultaneously. In this case press the “ALT” key on your keyboard together with the function-key “F4” Can also be a variable that replaces the word between < >
Start/Run...	Indicates a menu sequence. Here it reads Press the **Start** button and on the next menu select **Run**...

1.4 Revision history.

Revision number:	Description:
Revision 1.0 August 19, 2003	Initial release.
Revision 1.1 August 21, 2003	Some work on layout. No new stuff
Revision 1.2 September 4, 2003	Forgot to enter the addition of the `/etc/resolv.conf` file on the **I/A 50 Series** box. That's fixed now.
Revision 1.3 October 2, 2003	Some sites have changing domain name suffixes. Documented this with a possible modification to `.profile/go_dm.` Documented `/usr/fox/customer/hi/dmcfg` a little.
Revision 1.4 February 4, 2004	Layout conforms to Invensys standard. Some typos fixed.
Revision 1.5 March 15, 2005	Minor changes.

2 Dynamic IP Addressing and X-terminals

Setting up X-Terminals together with I/A 50 Series in combination with dedicated Display Managers or FoxViews, would traditionally involve a simple administration based on a "*hosts*" file. In that file, every PC and UNIX host would get a fixed IP address and the configuration of the Display Manager or FoxView behavior would be reasonably simple. The DM or Foxview would lookup the name for the corresponding IP address in the `/etc/hosts` file and take that name to lookup the dedicated DM or Foxview to use for that screen in the `/usr/fox/customer/hi/dmcfg` file. This type of administration is, in a larger network, difficult to maintain. A new **host** in the network would require all the separate "*hosts*" files be edited to reflect this change. However, there is a way to solve this isue and that is to make use of **DHCP** (**D**ynamic **H**ost **C**onfiguration **P**rotocol). This protocol allows hosts to get all required network parameters like the default gateway, default route, available **DNS** servers, domain name prefix etc. from one centrally maintained server making network maintenance a snap. On a few occasions some of our customers have at one time or another, expressed the wish to have all PC's in their network receive their IP addresses assigned from a **DHCP** server. This seems like the end of all problems but there is a catch: The Solaris 2.5.1 version used on Foxboro I/A 50 Series hosts does not support the use of **DHCP**.

Some ways to get around this problem:

1. Assign a fixed IP address to the Foxboro X-Terminals in the **DHCP** configuration.
2. Set the leasetime to a very long period that would, under normal conditions, never expire.
3. Setup an Domain Name System Server that interacts with a **DHCP** Server on the IT network, which will allow **I/A 50 Series** to resolve the IP address to a hostname.

We could waste your time and dive into the pro's and con's of all three options but this manual deals with option **3** as you may have guessed from the title :-)
Since one of the major problems is the **Solaris** version used in all **I/A 50 Series** which does not support **DHCP.** We must therefore take another approach to get things working. We had to find a way to make the **DHCP** information available to the **I/A 50 Series** host which at best can get info from a **DNS** server. The goal was to make configuring X-terminals and UNIX hosts as easy as possible preserving all the dedicated DM/Foxview functionality as we go along. The solution we opted for was to configure the Windows platform to get an IP address and additional configuration data from a **DHCP** server and make the **I/A 50 Series** UNIX host **DNS** aware.
This manual describes a solution that will allow you to have dynamic IP (and other) configuration settings, providing the information required by the host for normal operation. For the **DHCP** and **DNS** functions we opted for an open source solution based on **Mandrake Linux version 9.1 Download Edition** at the time of writing this manual.

2.1 Reference documents

Some of the paperwork you could get in to, trying to accomplish this:

- **X-Terminal Installation for Remote Display Manager and FoxView™**
  By Ron Deen, not partnumbered.
  Available from The Cassandra Project http://thecassandraproject.org/archive.html to name one

or http://home.hccnet.nl/r.deen/Docs to name another.

- **The DNS Howto – HOWTO become a totally small time DNS admin.**
  from Nicolai Langfeld, Jamie Norrish and others.
  Available from ftp://ftp.ibiblio.org/bub/Linux/docs/HOWTO/DNS-HOWTO
  A valuable HOWTO about maybe becoming something you don't want.
- **DNS and Bind 4th edition**
  ISBN: 0-596-00158-4.
  By Paul Albitz and Cricket Liu and available from O'Reilly & Associates Inc.
  The latest book covering BIND 9 and DNS Admin in general.

2.2 General requirements

To support dynamic IP addresses for the X-Terminals this manual assumes the following:

- The **I/A 50 Series** stations ALL have fixed IP addresses and must be running I/A Series version 4.3 or higher. I/A Series based on Solaris 8 was not a part of the test but may be working just fine.
- The **Windows XP Professional** based X-terminal obtains a valid IP address from a **DHCP** Server.
- A properly installed and configured **DHCP** Server is available on the secondary network.
- A properly installed and configured **DNS** Server is available on the secondary network and this **DNS** Server must be configured to be updated from **DHCP** automatically (it is a Dynamic DNS server).

For reference purposes we have included all the required files from the Linux based **DHCP** and **DNS** Server in this manual should you want to take this route.

2.3 The LAB Setup

In order to test our plans we used a small system containing an I/A 50 Series host, a Linux machine providing the DHCP and DNS servers we require for the test and a Windows PC for the X-Terminal functionality.

I/A 50 Series platform: a SPARCstation 5, 64Mb RAM, 2.1 Gb disk with I/A Series 4.3. (**Solaris 2.5.1**)
The Windows platform: a HP VL PC Pentium II 266, 160Mb Ram, on-board NIC, **Windows XP professional**.
The Linux Server platform: a Zenith Z-Station Pentium II 350MHz, 256Mb Ram, on-board NIC and additional NIC, running **Mandrake Linux 9.1 Download Edition** a 3CD package. Available at http://www.mandrakelinux.com/en in the download area. The **DNS** we used was BIND 9.2.2 which is maintained by the Internet Software Consortium. We installed it from the Mandrake CD's. The **DHCP** server is too maintained by the Internet Software Consortium and was also installed from the Mandrake CD's. The version on de CD was version 3.0–2pl2 Both Servers are covered under the BSD license.

3 System modifications per platform

In order to make everything work as intended, the three different platforms must be properly configured. This manual deals with all three of them.

1. Preparing the **Windows XP Professional** platform to be used as an X-Terminal. (section **4** on page **11**)
2. Setting up the **I/A 50 Series** platform which will be the host for the X-Terminals. (section **5** on page **19**)
3. The **Linux Mandrake** platform:
   - Installing and Configuring the **DHCP** Server. (section **6** on page **25**)
   - Installing and Configuring the **DNS** Server. (section **7** on page **31**)

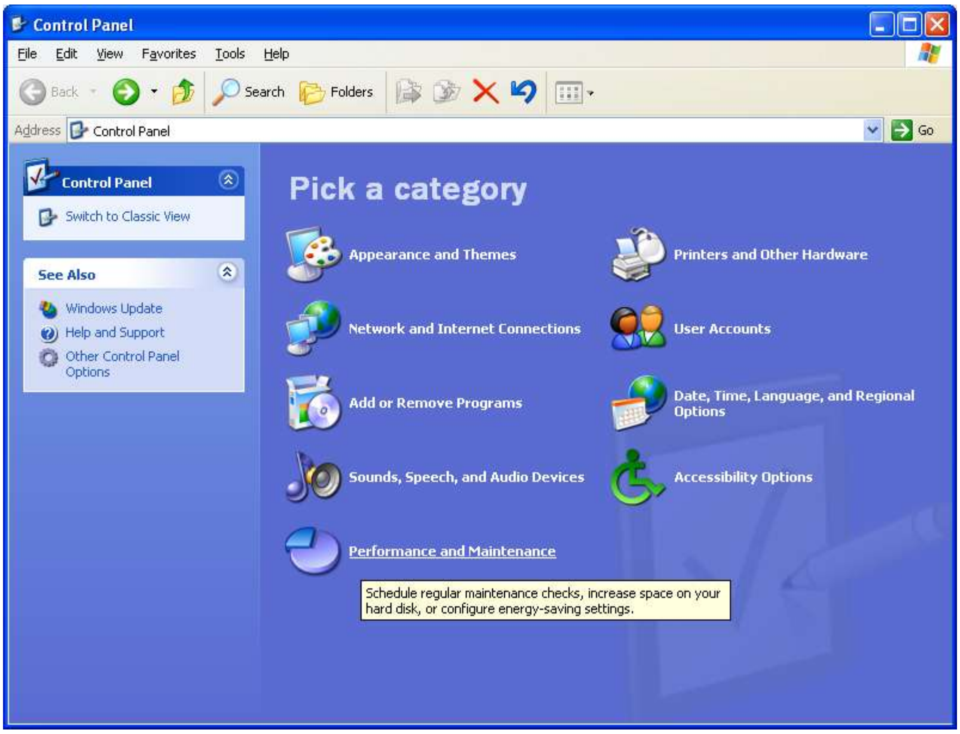
Have a look at the sections relevant to the task. You can skip the sections dealing with **DNS** and **DHCP** in total if you are sure that these services are setup correct.

4 Preparing the Windows XP Platform

The Windows platform must be configured to obtain an IP address from a **DHCP** Server. How this is configured you will find here. Other settings that are important to this platform like the default gateway, netmask, the **DNS** Server address, etc will be provided by the **DHCP** Server, so we don't worry about these things here. Although this may be very straightforward stuff for you (of course you have performed this trick at least a dozen times for Windows NT and 95) we have provided the screen-shots for XP since I remember being utterly lost on this platform when I wanted to do exactly the same things I had done before...

4.1 Setting the Windows XP machine hostname.

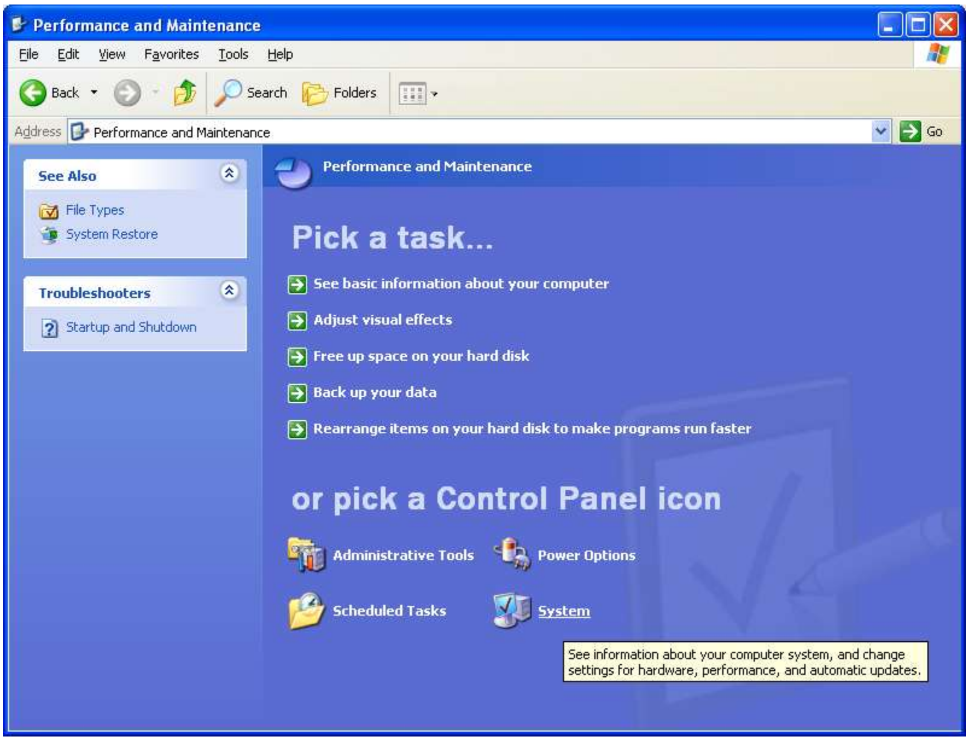
One of the things to do is identify the PC's *hostname*. Dedicated Foxview and DM's are related to the PC name to check whether you are a valid user or not. To find out and/or set this machine name we should have a look under the hood and that would be: Choose the **Control Panel** from the **Start** menu... (**Start/Control Panel**)



Schedule regular maintenance checks, increase space on your hard disk, or configure energy-saving settings.

Illustration 1 The Control Panel (Performance and Maintenance)

Choose the **Performance and Maintenance** category as shown above and click on that. The next screen will show you the options you have here.


Performance and Maintenance
File Edit View Favorites Tools Help
Back Search Folders
Address Performance and Maintenance Go
See Also
File Types
System Restore
Troubleshooters
Startup and Shutdown
Performance and Maintenance

Pick a task...

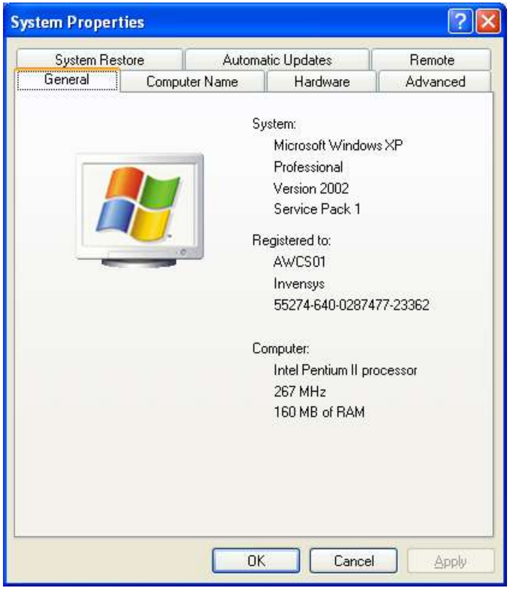
- See basic information about your computer
- Adjust visual effects
- Free up space on your hard disk
- Back up your data
- Rearrange items on your hard disk to make programs run faster

or pick a Control Panel icon

See information about your computer system, and change settings for hardware, performance, and automatic updates.

Illustration 2 Control Panel > Performance and Maintenance > System

We want to have have a peek in one of the **Control Panel** icons. For our task we need to select the **System** icon which brings you this:



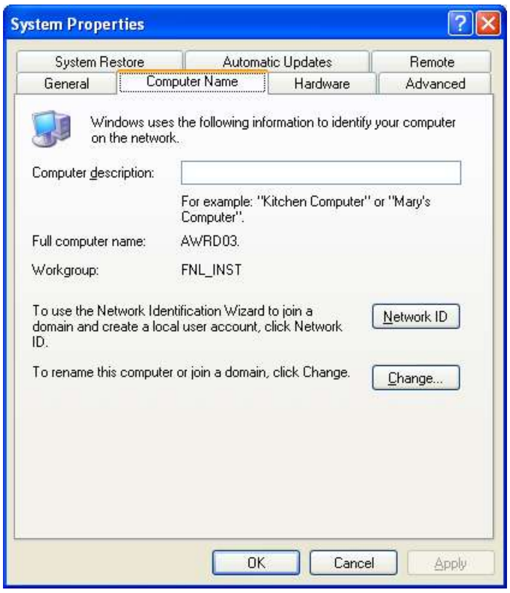
System:
Microsoft Windows XP
Professional
Version 2002
Service Pack 1

Registered to:
AWCS01
Invensys
55274-640-0287477-23362

Computer:
Intel Pentium II processor
267 MHz
160 MB of RAM

Illustration 3 System Properties (General tab selected)

Activate the tab that reads **Computer Name** to proceed...



Windows uses the following information to identify your computer on the network.

Computer description:

For example: "Kitchen Computer" or "Mary's Computer".

Full computer name: AWRD03.

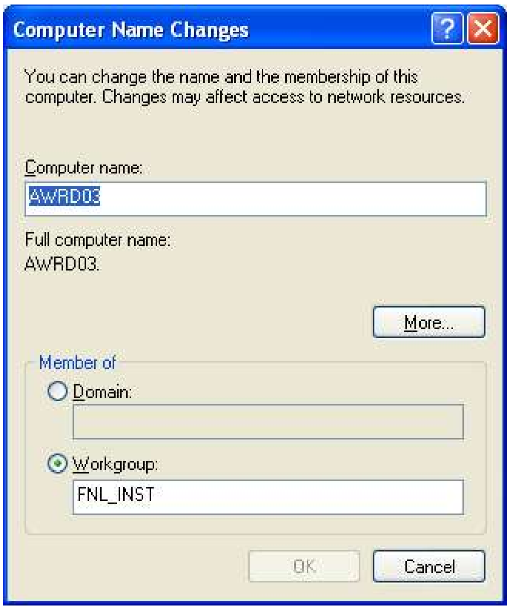
Workgroup: FNL_INST

To use the Network Identification Wizard to join a domain and create a local user account, click Network ID.

To rename this computer or join a domain, click Change.

Illustration 4 System Properties (Computer Name tab selected)

Here you can set the PC name on the network (i.e: the *hostname*). Please note that it is this name that will be used to identify your PC on the I/A 50 Series hosts. On these hosts, the file /**usr/fox/customer/hi/dmcfg** is used to map a dedicated Display Manager or FoxView to a specific machine name. The **Computer Name** should match the one in the **dmcfg** file. Choose the **Change** button to make the modifications you desire: In this example we used the name **AWRD03** because we had a dedicated DM mapped to that name.



You can change the name and the membership of this computer. Changes may affect access to network resources.

Computer name: AWRD03

Full computer name:
AWRD03.

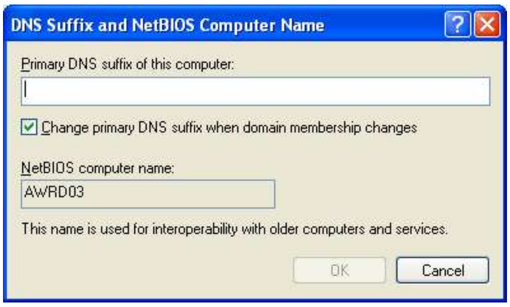
Member of

○ Domain:

● Workgroup: FNL_INST

Illustration 5 Changing the Computer name

Follow the **More...** button to make one more additional change:



DNS Suffix and NetBIOS Computer Name

Primary DNS suffix of this computer:

☑ Change primary DNS suffix when domain membership changes

NetBIOS computer name:
AWRD03

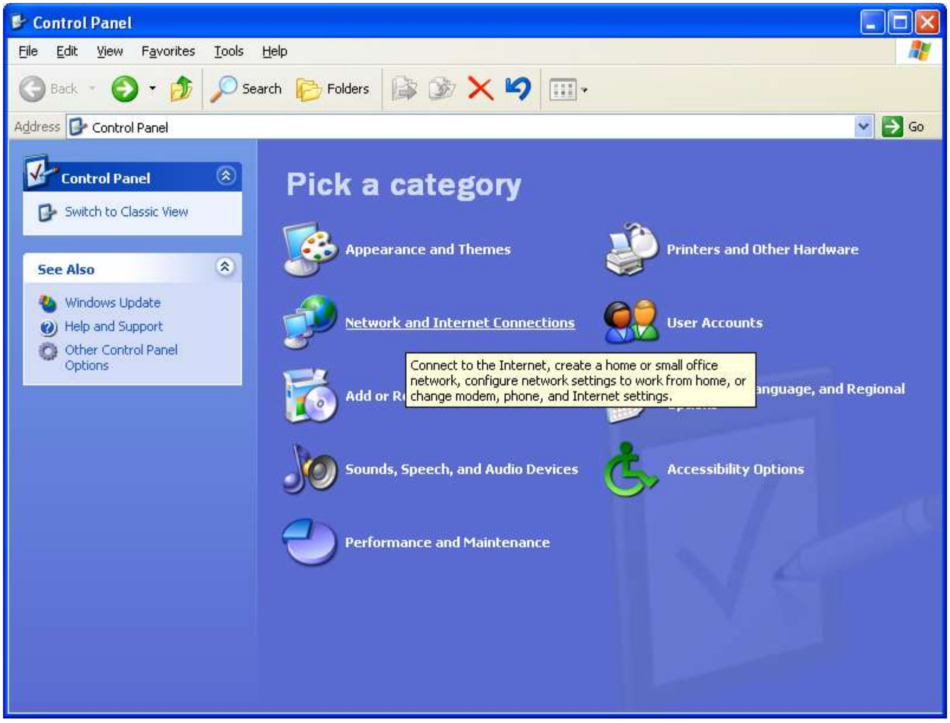
This name is used for interoperability with older computers and services.

Illustration 6 DNS Suffix and NetBIOS Computer name

Check the box next to **Change primary DNS suffix when domain membership changes.**

4.2 The TCP/IP related settings

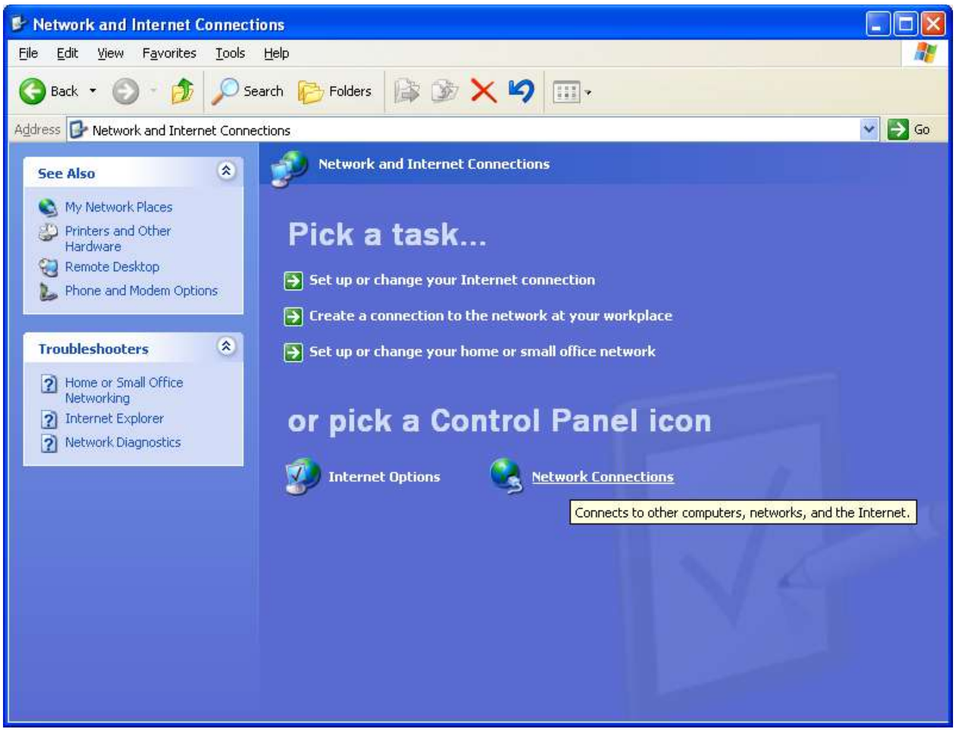
To set the TCP/IP protocol and **DNS** settings correct for this setup again pick the **Control Panel**. From the **Control Panel** select the **Network and Internet Connections** category.



Connect to the Internet, create a home or small office network, configure network settings to work from home, or change modem, phone, and Internet settings.

Illustration 7 Control Panel (Network and Internet Connections)

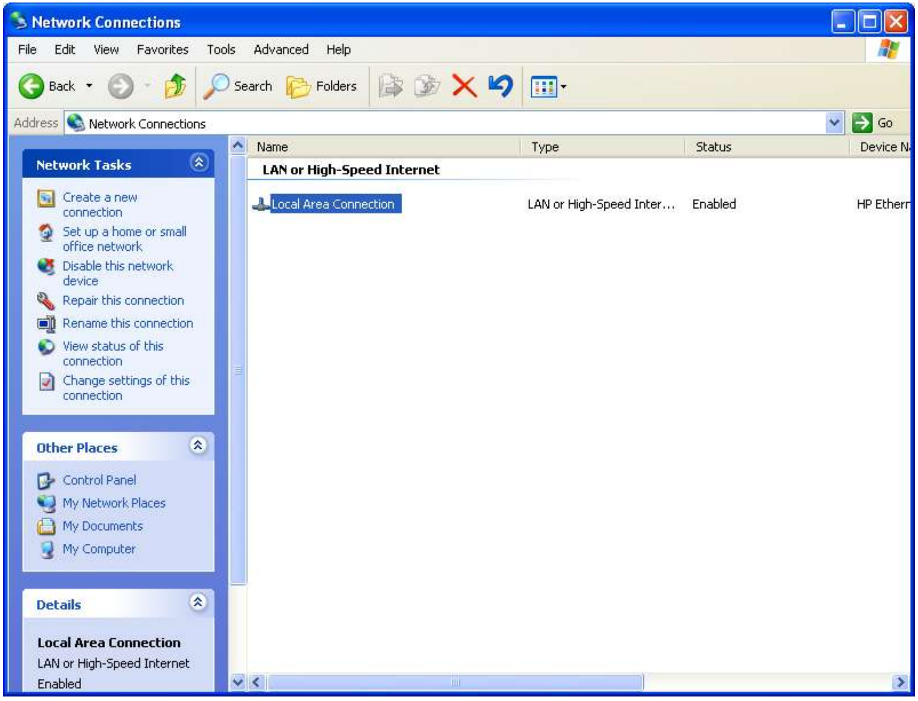
When the screen as below appears, select the icon **Network Connections .**


Network and Internet Connections
File Edit View Favorites Tools Help
Back Search Folders
Address Network and Internet Connections Go
See Also
My Network Places
Printers and Other Hardware
Remote Desktop
Phone and Modem Options
Troubleshooters
Home or Small Office Networking
Internet Explorer
Network Diagnostics
Network and Internet Connections
Pick a task...
Set up or change your Internet connection
Create a connection to the network at your workplace
Set up or change your home or small office network
or pick a Control Panel icon
Internet Options
Network Connections

Connects to other computers, networks, and the Internet.

Illustration 8 Control Panel > Network Connections

When you have done that, you get the screen similar to the one below:

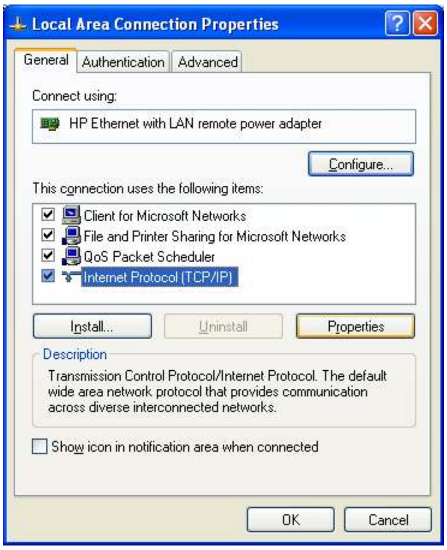

Network Connections
File Edit View Favorites Tools Advanced Help
Back Search Folders
Address Network Connections Go
Network Tasks
Create a new connection
Set up a home or small office network
Disable this network device
Repair this connection
Rename this connection
View status of this connection
Change settings of this connection
Other Places
Control Panel
My Network Places
My Documents
My Computer
Details

Name	Type	Status	Device N
LAN or High-Speed Internet			
Local Area Connection	LAN or High-Speed Inter…	Enabled	HP Etherr

Local Area Connection
LAN or High-Speed Internet
Enabled

Illustration 9 Network Connections, Local Area Connection selected

Next, select the **Local Area Connection** , "*right click*"on that line and choose **Properties,** you get a screen similar to the one shown here:



Local Area Connection Properties

Connect using:

HP Ethernet with LAN remote power adapter

This connection uses the following items:

☑ Client for Microsoft Networks
☑ File and Printer Sharing for Microsoft Networks
☑ QoS Packet Scheduler
☑ Internet Protocol (TCP/IP)

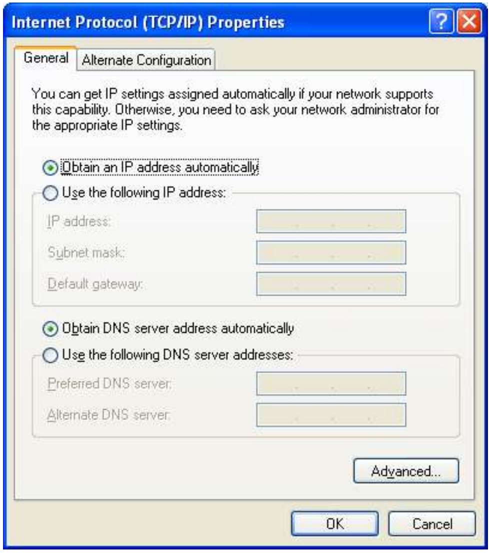
Description

Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

□ Show icon in notification area when connected

Illustration 10 Local Area Connection Properties

Again select the **Internet Protocol (TCP/IP)** line and click on the **Properties** button. On the **General** TAB some modifications are required:



Internet Protocol (TCP/IP) Properties

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

● Obtain an IP address automatically
○ Use the following IP address:
IP address: . . .
Subnet mask: . . .
Default gateway: . . .

● Obtain DNS server address automatically
○ Use the following DNS server addresses:
Preferred DNS server: . . .
Alternate DNS server: . . .

Illustration 11 Internet Protocol (TCP/IP) Properties

Make sure that the selections are matching those in the example on screen i.e.:

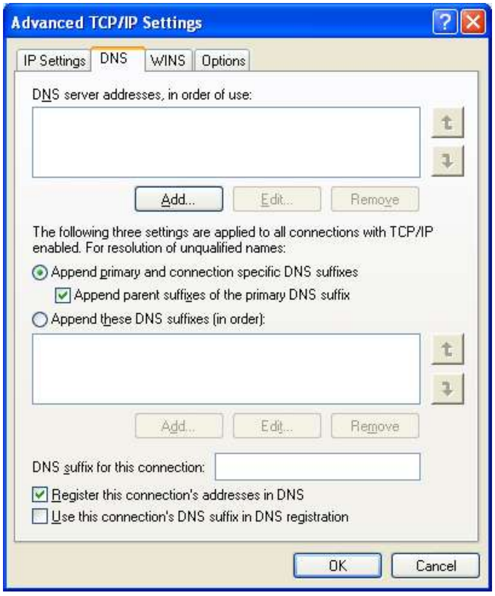
- **Obtain an IP address automatically** (this will enable **DHCP** for this Windows client) and
- **Obtain DNS server address automatically** (unless you get other info from your System administrator).

and when this is done, pick the **Advanced...** button.

In the next screen the **DNS** tab was chosen. Make the changes as shown, i.e: check the boxes that read:

- **Append primary and connection specific DNS suffixes**
- **Append parent suffixes of the primary DNS suffix** and

- **Register this connection's addresses in DNS** (on by default)



DNS server addresses, in order of use:

The following three settings are applied to all connections with TCP/IP enabled. For resolution of unqualified names:

● Append primary and connection specific DNS suffixes

☑ Append parent suffixes of the primary DNS suffix

○ Append these DNS suffixes (in order):

DNS suffix for this connection:

☑ Register this connection's addresses in DNS

□ Use this connection's DNS suffix in DNS registration

Illustration 12 Advanced TCP/IP Settings (DNS tab selected)

This concludes the modifications required on the **Windows XP Professional** platform. **The I/A 50 Series** platform comes next.

5 Setting up the I/A 50 Series platform

The modifications on the **I/A 50 Series** platform are fairly minimal. We must make sure that the UNIX host knows that there is a **DNS** server available on the network and the host should place it's **DNS** queries there in case the request can't be resolved locally. This implies making modifications to the `/etc/hosts` file, the `/etc/nsswitch` file and the `/etc/resolv.conf` file on the UNIX host. In this section you can also find some information about the Display Manager Configuration file: `/usr/fox/customer/hi/dmcfg` on the X-Terminal host.

5.1 The `/etc/hosts` table

The `/etc/hosts` file would normally hold all IP address and name data for the X-terminals and other hosts in your system. This file could be almost identical on all hosts and should be maintained properly. Since we now will have a centralized place where this information is stored we can get away with a much simpler `/etc/hosts` file. For starters, any references to your PC X-Terminal based hardware can be removed from this file on your UNIX hosts. This means that the *host file* can almost be the one that is created by the **System Definitions** program. An example from our LAB test setup, note that only the second Ethernet port is declared in here:

[AW51RD /etc/hosts]

```
#
# Internet host table
#
127.0.0.1       localhost       loghost hw1197
192.168.1.2     AW51P2
#******************************************************
# Start of I/A hosts
# created Mon Jul 7 10:27:30 GMT 2003
#******************************************************
#
# The following host entries were created by the I/A
# Software Install sub-system. Any additional entries
# should be placed AFTER the End delimiter.
#
151.128.16.65   AW51RD
#
#******************************************************
# End of I/A hosts
#******************************************************
AW51RD#
```

After we have cleaned up the `hosts` file there is one other file to edit. This is the Name Service Switch configuration file and that's coming next…

5.2 The `/etc/nsswitch.conf` file

In order to make the UNIX host look further than the default `hosts` file (which won't tell us too much anymore) we must make a small change to the file that takes care of this for us. The file for this is located in the `/etc` directory and is named `nsswitch.conf`. This configuration file takes care of the **Name Service Switch**. (Who says that filenames in UNIX don't make sense?). In this file, the place to look for is the line that starts with "`hosts:`", usually this is followed by the word "`files`" indicating that for *hosts* resolving only the *hosts file* will be used. We want to make sure that if that fails (and it will) the system should try to find the **DNS** server and get the required information there. For this the word "**`dns`**" must be added to the line after "`file`" as shown in the example.

[/etc/nsswitch.conf]

```
::::::::::::::
nsswitch.conf
::::::::::::::
#
# /etc/nsswitch.files:
#
# An example file that could be copied over to /etc/nsswitch.conf; it
# does not use any naming service.
#
# "hosts:" and "services:" in this file are used only if the
# /etc/netconfig file has a "-" for nametoaddr_libs of "inet" transports.

passwd:     files
group:      files
hosts:      files dns
networks:   files
protocols:  files
rpc:        files
ethers:     files
netmasks:   files
bootparams: files
publickey:  files
# At present there isn't a 'files' backend for netgroup;  the system will#
figure it out pretty quickly, and won't use netgroups at all.
netgroup:   files
automount:  files
aliases:    files
services:   files
sendmailvars:   files
```

5.3 The `/etc/resolv.conf` file

The changes we have made until now prepare the UNIX host to query a DNS server for name lookups. However, the system still does not know where to find such a DNS Server. Therefore one last change is required and that is the file `/etc/resolv.conf`. This file is not found on standard **I/A 50 Series** stations and must be added. The function of this file is to make the **I/A 50 Series** station do its resolving starting in the local *domain* and when that fails, consult the *nameserer(s)* found is this file to see if the IP/hostname resolving can be accomplished there..This file contains the *domain name* we are part of and the *IP address* for the **DNS** server to be consulted.

[AW51RD /etc/resolv.conf]

```
domain     domain.org
nameserver 192.168.1.1
```

When you will do an `nslookup hostname` using the example `resolv.conf` file above, because of the *domain* entry here, the *hostname* will be appended by the *domain* given here. I.e.: `hostname` will become `hostname.domain.org` in this example. When however you enter: `hostname.pookie.net` the name will remain unchanged since you did provide domain information.

5.4 The `dmcfg` file

This file is used to configure the **Display Manager/FoxView** defaults. Here you define which (if any) DM will be assigned to which X-Terminal. In a little example file we will explain the setup for this where we define one DM to one specific DM and one DM to a group of X-Terminals where only the first two will get one. As you may know already, the `dmcfg` file is found in the directory `/usr/fox/customer/hi` on your host AW of WP and the example shown is derived from one that was created by the **Display and Alarm Manager Configurator**:

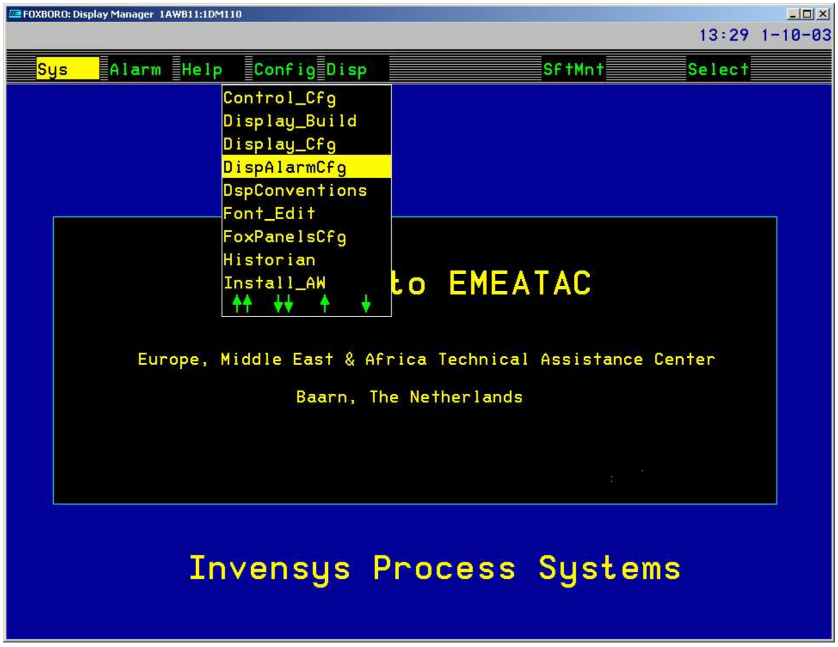


Illustration 13 Invoking the GUI Display and Alarm Manager Configurator

Although we won't be using the **Display and Alarm Manager Configurator** here, the example we start off with is one that was created with this configurator. The biggest advantage of using a `dmcfg` that was created this way lies in the fact that it is much easier to read because all comments are stripped from it and it contains only relevant data. Here is the example:

```
#--------------------------------------------------------------
# "dmcfg"  Tue Sep 23 09:44:50 GMT 2003
#--------------------------------------------------------------
SCREEN  SUPPORT   pc-1-supt pc-2-supt pc-3-supt
SCREEN  SERVICE   pc-1-serv pc-2-serv pc-3-serv pc-3-serv pc-4-serv pc-5-serv pc-
6-serv pc-7-serv
SCREEN  OPERATION pc-1-ccr pc-2-ccr pc-3-ccr pc-4-ccr pc-5-ccr pc-6-ccr pc-7-ccr
pc-8-ccr pc-9-ccr
#--------------------------------------------------------------
TYPE    foxDefault      DSM     O  -
TYPE    VIEW_ONLY       DSM     V  Init_Env
TYPE    _DEFAULT        DSM     O  Proc_Op_Env
#--------------------------------------------------------------
AMTYPE  foxDefault              O  -
AMTYPE  foxFullScreen           O  -
AMTYPE  foxFullSq               O  -
AMTYPE  foxGrids                O  -
AMTYPE  foxOldCAD               O  -
AMTYPE  foxQuarter              O  -
AMTYPE  foxStrips               O  -
AMTYPE  _DEFAULT                O  Proc_Op_Env
#--------------------------------------------------------------
NAME    OMAW02  DM0000  $SUPPORT        foxDefault
NAME    OMAW02  DM0001  $OPERATION      VIEW_ONLY
NAME    OMAW02  DM0002  $SERVICE        VIEW_ONLY
NAME    OMAW02  DM0003  $SERVICE        VIEW_ONLY
NAME    OMAW02  DM0004  $SERVICE        VIEW_ONLY
NAME    OMAW02  DM0005  $SUPPORT        foxDefault
NAME    OMAW02  DM0006  AWRD03          foxDefault
#--------------------------------------------------------------
AMNAME  OMAW02  AM0000  -               foxDefault
#--------------------------------------------------------------
```

In the example shown above:
Two DM's are dedicated to the PC's from the support group, and for that reason a **SCREEN** variable was declared containing the PC names from the Support group that should have access to one of the two DM's.
For the Service group too, a **SCREEN** variable was created with the PC names from their systems. This group shares three DM's.
The Operators share one DM license among them. (I just hope they have some WP's in addition to this).
One additional DM was created that is dedicated to one specific X-Terminal, one DM is assigned to the PC X-Terminal with name **AWRD03**.

5.4.1 Some additional notes

Some things to keep in mind when editing the `dmcfg` file:

- PC names as shown in the `dmcfg` file are *case sensitive*. E.g. when you add a PC with name **xterm1** and the `dmcfg` file shows **XTERM1,** the message: **No DM configured for this screen** will appear with the obvious consequence you will not get one.
- Adding PC's to a `SCREEN` list can not be done from the GUI configurator. You must make the changes to the actual `dmcfg` file using a text editor like *vi*.
- When you start an X-Terminal session, the `go_dm/.profile` file will attempt to set the **DISPLAY** variable. This variable will contain the **Fully Qualified Domain Name** for your PC. It is exactly this name that should be set in the `dmcfg` file. So, to find out what the **DNS** has in store for you try to do an **`nslookup <pcname>`** from the prompt and see what the response is:

  ```
  # nslookup xterm1<cr>
  Name:     xterm1.domain.org
  Address:   10.31.16.116
  ```

  When you get a result similar to the one above and the DM does not work, try to add PC name for the X-Terminal as displayed in the response, i.e.: **xterm1.domain.org** so you will get as line that looks something like**:**

  ```
  NAME   OMAW02   DM0005   xterm1.domain.org   FoxDefault
  ```

5.4.2 Filtering the hostname from your FQDN

When you have the scenario where you must always enter the FQDN (the abbriviation for **Fully Qualified Domain Name**) the in the `dmcfg` file **and** this is not as desired because your domain changes regularly, it may be an idea to strip everything away from the FQDN and leave the host portion only. This can be achieved by making a small modification to the `.profile/go_dm` file located in the `/opt/ia` directory. Below you will find a little part of that file with **(`in bold and red print`)** the changes suggested:

```
SIGUSR1=`grep SIGUSR1 $SIG_PATH | awk '{ printf ( "%s", $3 ) }'`

REM_STATION=`who am i| awk '{print $6}' | tr -d "()" | awk -F. '{print $1}'`
LOC_STATION=`uname -n`
```

```
DISPLAY=$REM_STATION:0 ; export DISPLAY
echo Setting DISPLAY to $DISPLAY
```

Without the modification the **DISPLAY** variable is filled with **`HOSTNAME.domain.org:0`** (i.e.:the FQDN) while the modified file will produce the **DISPLAY** variable with only **`HOSTNAME:0`**.

These are all the changes required on the **I/A 50 Series** platform.

6 Setting up the DHCP Server

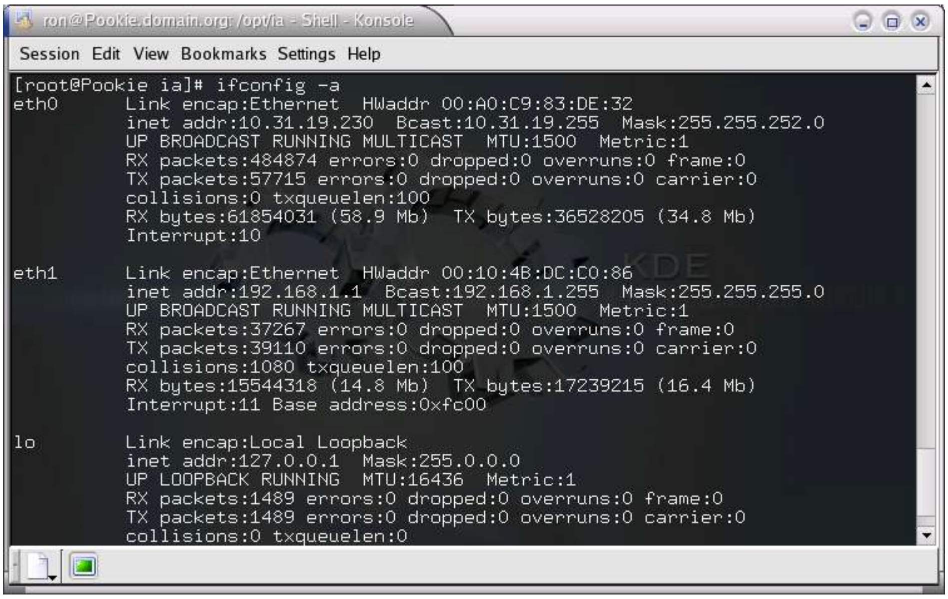
In order for our setup to work, we have some special requirements regarding the **DHCP**/**DNS** combo. The most important requirement is that **DNS** is updated automatically by the **DHCP** server regarding information that is handed out to clients. Because the IP addresses are provided to PC clients on the network by **DHCP**, it would be a requirement that this information was propagated to the **DNS** server since we are using an *nslookup* to the **DNS** server to get a name that matches an IP address. The latest versions of **DHCP** and **DNS** do support this functionality but finding the information on how to achieve this was another story. Nevertheless, if your plant already has a **DHCP**/**DNS** combination that meets this requirement, you already have completed the setup. In this chapter we will configure a **DHCP** Server that meets our requirements. **This setup was tested using a PC with Mandrake Linux version 9.1 download edition.** Additionally we added the **DHCP** server to this configuration. Although there is no need to use Mandrake or any other Open Source initiative or this purpose, this section hopefully will provide some information about the requirements for the **DHCP** server we used. As mentioned here, Linux is used for this setup. The reason beeing it comes without any licensing hassle and is well documented and supported.

6.1 The Server configuration

Before we start, a few thing are assumed to be in working order:

- You have the desired network interface (eth1 in this case) configured and up and running.
- A ping from from the "outside" to that interface will be successfull.
- The output from `ifconfig` will something similar to the example below.
- You can *telnet* and *ping* the server via the *localhost* name.

Let's have a look at the hardware we started with, more specific the network interfaces installed. The Server was equipped with two network interfaces: one on-board and one added as a PCI card. The settings used can be obtained using the command "`ifconfig -a`" from a terminal window:


ron@Pookie.domain.org: /opt/ia - Shell - Konsole
Session Edit View Bookmarks Settings Help

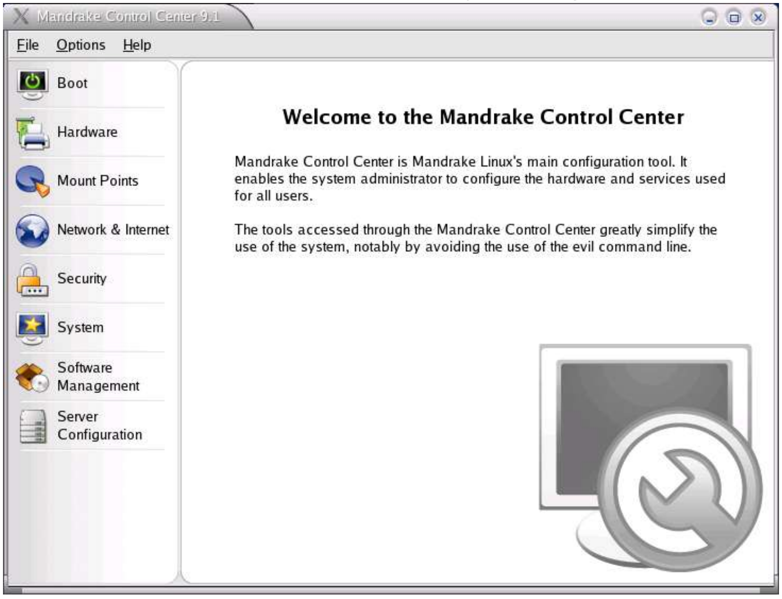
```
[root@Pookie ia]# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:A0:C9:83:DE:32
          inet addr:10.31.19.230  Bcast:10.31.19.255  Mask:255.255.252.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:484874 errors:0 dropped:0 overruns:0 frame:0
          TX packets:57715 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:61854031 (58.9 Mb)  TX bytes:36528205 (34.8 Mb)
          Interrupt:10

eth1      Link encap:Ethernet  HWaddr 00:10:4B:DC:C0:86
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:37267 errors:0 dropped:0 overruns:0 frame:0
          TX packets:39110 errors:0 dropped:0 overruns:0 carrier:0
          collisions:1080 txqueuelen:100
          RX bytes:15544318 (14.8 Mb)  TX bytes:17239215 (16.4 Mb)
          Interrupt:11 Base address:0xfc00

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1489 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1489 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
```

Illustration 14 On the Linux box: results from "ifconfig –a" command

As we can see (we are primarily interested in the **eth1** card), the network is configured with IP address **192.168.1.1/255.255.255.0**. (Note the hardware address as we will use that in the `dhcpd.conf` file to force that IP address to this interface). The **eth0** network interface was, in our configuration, connected to the office network. We used that to obtain data from the outside world. It is of no real importance to our task here.

6.2 Installing DHCP with the Mandrake Control Center

The place to install your software is the **Mandrake Control Center**. It is usually selectable from the bar at the buttom of the screen or by invoking the command "**drakconf**" from a terminal:



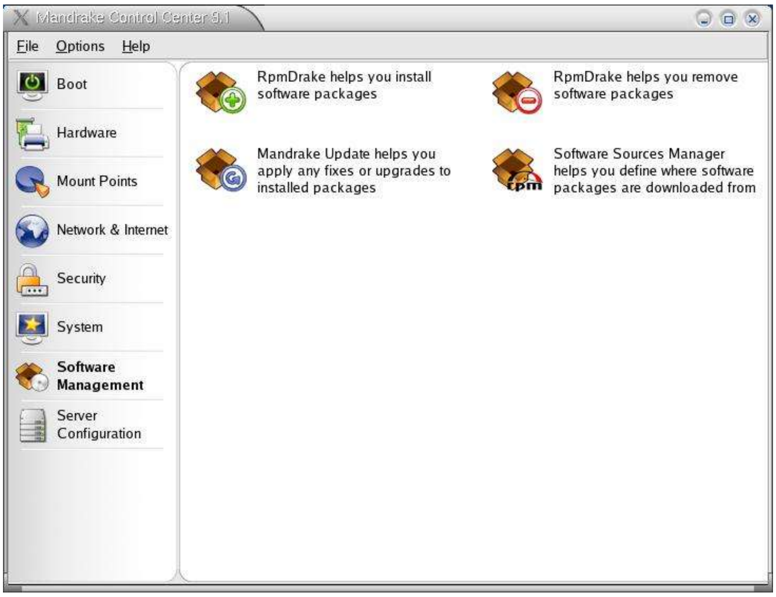
Welcome to the Mandrake Control Center

Mandrake Control Center is Mandrake Linux's main configuration tool. It enables the system administrator to configure the hardware and services used for all users.

The tools accessed through the Mandrake Control Center greatly simplify the use of the system, notably by avoiding the use of the evil command line.

Illustration 15 The Mandrake Control Center

We will be needing the **Software Management** entry from this screen to install the **DHCP** Server (if not already installed that is). You will get a screen as below where you click on the **RpmDrake** icon to install the software we need:



RpmDrake helps you install software packages

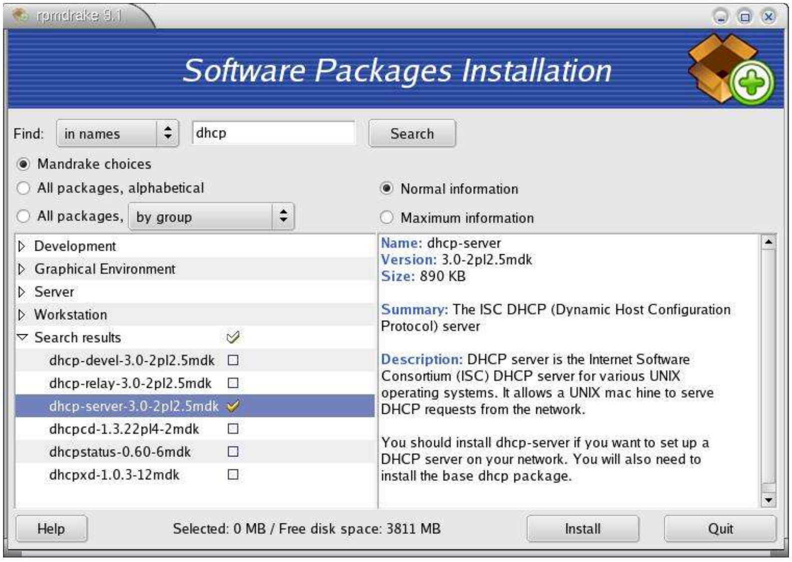
RpmDrake helps you remove software packages

Mandrake Update helps you apply any fixes or upgrades to installed packages

Software Sources Manager helps you define where software packages are downloaded from

Illustration 16 Mandrake Software Management screen

You now have a display as shown below which will allow you to enter a search item and present you with the options. In our example we want to install the **DHCP** server, so let's search for that:



Find: in names dhcp

● Mandrake choices
○ All packages, alphabetical
○ All packages, by group

● Normal information
○ Maximum information

- Development
- Graphical Environment
- Server
- Workstation
- Search results ✓
  - dhcp-devel-3.0-2pl2.5mdk □
  - dhcp-relay-3.0-2pl2.5mdk □
  - dhcp-server-3.0-2pl2.5mdk ✓
  - dhcpcd-1.3.22pl4-2mdk □
  - dhcpstatus-0.60-6mdk □
  - dhcpxd-1.0.3-12mdk □

Name: dhcp-server
Version: 3.0-2pl2.5mdk
Size: 890 KB

Summary: The ISC DHCP (Dynamic Host Configuration Protocol) server

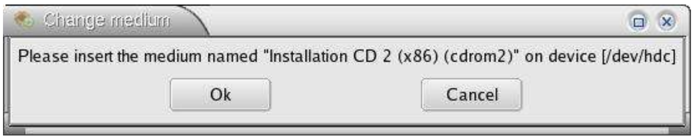
Description: DHCP server is the Internet Software Consortium (ISC) DHCP server for various UNIX operating systems. It allows a UNIX mac hine to serve DHCP requests from the network.

You should install dhcp-server if you want to set up a DHCP server on your network. You will also need to install the base dhcp package.

Selected: 0 MB / Free disk space: 3811 MB

Illustration 17 The dhcp search results (selection made)

As we can see, the search resulted in several hits where we selected the `dhcp-server-3.0-2pl2.5mdk` package. This package is all we need from the listed options. When we click the **Install** button now,you are prompted to insert the **Installation CD 2**. Follow the directions on screen to complete the installation and you are done. You should have **CD 2** at hand of course:



Please insert the medium named "Installation CD 2 (x86) (cdrom2)" on device [/dev/hdc]

Illustration 18 Insert the correct CD for the job

6.3 The DHCP Server configuration

After we have installed the **DHCP** server, we can continue with the configuration. The **DHCP** configuration file is quite simple. The file is printed here and we will try to explain the options used. The **DHCP** configuration file is usually found in the ”`/etc`“ directory and is named “**`dhcpd.conf`**“. The contents for our configuration file are printed below:

[/etc/dhcpd.conf]

```
ddns-update-style interim;
subnet 192.168.1.0 netmask 255.255.255.0 {
        # default gateway
        option routers 192.168.1.1;
        option subnet-mask 255.255.255.0;
        option broadcast-address 192.168.1.255;
        option domain-name-servers 192.168.1.1;
        option domain-name "domain.org";

        range dynamic-bootp 192.168.1.12 192.168.1.25;
        default-lease-time 3600;
        max-lease-time 7200;

        # we want the nameserver to appear at a fixed address
        host ns {
                next-server pookie.domain.org;
                hardware ethernet 00:10:4B:DC:C0:86;
                fixed-address 192.168.1.1;
        }
}
```

This file, once you get used to it, is reasonably straightforward. (Yes, strange things happen to you when look at weird things long enough...) Some of the options and what they mean:

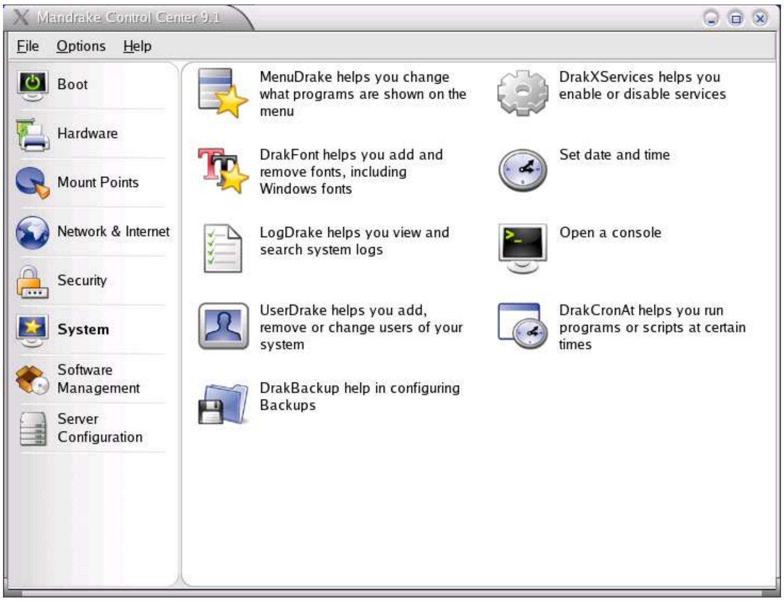
- **`ddns-update-style interim`**
  Allows for dynamic updates for the **DNS** Server.
- **`subnet`**
  Defines the subnet mask for the global network.
- **`option routers`**
  Sets the default router for the clients.
- **`option subnet-mask`**
  Tells the clients to use this subnet mask.
- **`option broadcast-address`**
  Sets the broadcast address to this value for all clients.
- **`option domain-name-servers`**
  Informs the clients about their **DNS** server IP.
- **`option domain-name`**
  Contains the domain name for the clients.
- **`range dynamic-bootp`**
  Specifies the IP range to be handed out to requesting clients.
- **`default-lease-time`**
  If a client makes no special request, specifies the lease time after which the IP address expires.
- **`max-lease-time`**
  If a clients request a specific lease time, sets the max to this value.

- **`host ns`**
  The host *name server* gets some options too. In this case it gets a fixed ip address from **DHCP**.

The required configuration work for the **DHCP** Server is now completed.

6.3.1 Starting the DHCP server

DHCP will not start just by force of will. On a Mandrake system, the Mandrake Control Center is the way to configure the Server startup. After selecting the **System** button on the left, choose the icon next to **DrakXServices:**



MenuDrake helps you change what programs are shown on the menu

DrakXServices helps you enable or disable services

DrakFont helps you add and remove fonts, including Windows fonts

Set date and time

LogDrake helps you view and search system logs

Open a console

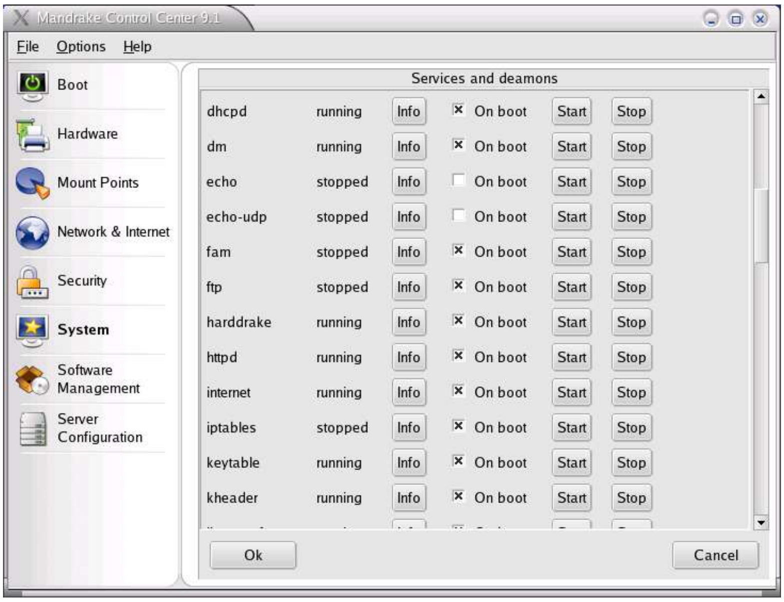
UserDrake helps you add, remove or change users of your system

DrakCronAt helps you run programs or scripts at certain times

DrakBackup help in configuring Backups

Illustration 19 Finding the Services to start via DraxServices

You then get a list of available services on your system. They are listed in alphabetical order. Scroll down until you see the "*dhcpd*" program and select the "**On boot**" button and try to start the server. The picture below shows the "*dhcpd*" (the **DHCP** daemon) is running and the "**On boot**" button is checked.

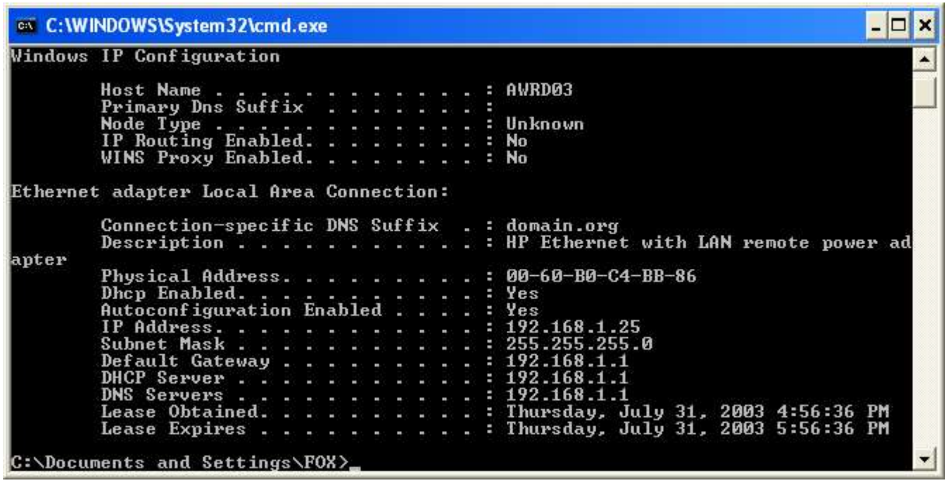


Services and deamons

dhcpd	running	Info	☒ On boot	Start	Stop
dm	running	Info	☒ On boot	Start	Stop
echo	stopped	Info	□ On boot	Start	Stop
echo-udp	stopped	Info	□ On boot	Start	Stop
fam	stopped	Info	☒ On boot	Start	Stop
ftp	stopped	Info	☒ On boot	Start	Stop
harddrake	running	Info	☒ On boot	Start	Stop
httpd	running	Info	☒ On boot	Start	Stop
internet	running	Info	☒ On boot	Start	Stop
iptables	stopped	Info	☒ On boot	Start	Stop
keytable	running	Info	☒ On boot	Start	Stop
kheader	running	Info	☒ On boot	Start	Stop

Illustration 20 Verifying if the DHCP daemon is running and startup configured correctly.

When the server is started, a client requesting an IP address should receive the settings as set in the `/etc/dhcpd.conf` file we configured earlier. This can be checked after the **Windows XP** client was booted by issuing the command "`ipconfig /all`" from any **cmd** command box. The response must be similar to the one shown here:



```
Windows IP Configuration

        Host Name . . . . . . . . . . . . : AWRD03
        Primary Dns Suffix  . . . . . . . :
        Node Type . . . . . . . . . . . . : Unknown
        IP Routing Enabled. . . . . . . . : No
        WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . : domain.org
        Description . . . . . . . . . . . : HP Ethernet with LAN remote power ad
apter
        Physical Address. . . . . . . . . : 00-60-B0-C4-BB-86
        Dhcp Enabled. . . . . . . . . . . : Yes
        Autoconfiguration Enabled . . . . : Yes
        IP Address. . . . . . . . . . . . : 192.168.1.25
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.1.1
        DHCP Server . . . . . . . . . . . : 192.168.1.1
        DNS Servers . . . . . . . . . . . : 192.168.1.1
        Lease Obtained. . . . . . . . . . : Thursday, July 31, 2003 4:56:36 PM
        Lease Expires . . . . . . . . . . : Thursday, July 31, 2003 5:56:36 PM

C:\Documents and Settings\FOX>_
```

Illustration 21 Windows XP: output from "ifconfig /all" command.

If you get a similar response, we are ready to proceed with the next step: Setting up your **DNS.**

7 Setting up the DNS Server

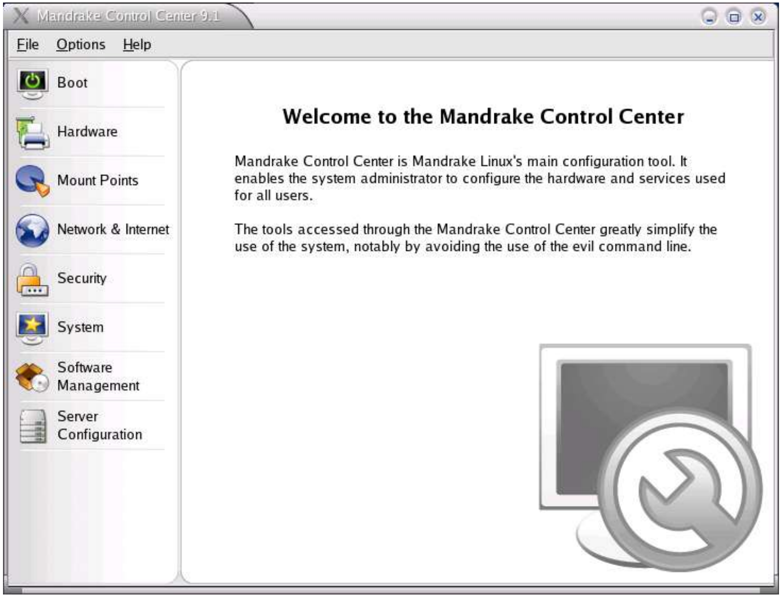
A successful **DNS** Server setup can be a little harder to accomplish. If this is what you have to do, you may dig into some related documentation as mentioned in the beginning of this manual, consult the Internet or find someone else to do it ;-) **DNS**, which is the abbreviation for **Domain Name System**, is a system that provides the mapping between IP addresses to names and vice versa. This can be used to look up an IP address corresponding to a name, or find a name to an IP address. The latter part is our main interest here since we know an IP address (the X-Terminal received that from the **DHCP** Server) and want to know the name that goes with it.

7.1 Confusing: BIND?, DNS?, named?

One of the first things we may need to clear out is the different names you will find in this section. We are going to setup a ***DNS Server***. This requires a package called ***BIND9*** (or BIND8 which is a little older). This package contains the actual "program" that will do the work for you, but this program is called ***named*** *(for **name d**aemon)*. **BIND** (Berkely Internet Name Domain Software) is one of the most common implementations of **DNS** which implicates that ***BIND** is **DNS*** but ***DNS** is not limited to **BIND***, there are others. In this setup we will restrict ourselves to the Berkely implementation. So ***BIND*** is the ***DNS*** Server implementation we use here and that starts a daemon that's called ***named***, OK? All three names are used in the related documentation. We installed the **BIND9** package from the **Mandrake 9.1 Download Edition** CD-Roms on our Linux machine, (it is on one the three download CD's so there should be no problem getting that part done. After the installation we can have a look at what we have.

7.2 Installing BIND (DNS) with the Mandrake Control Center

The place to install your software is the **Mandrake Control Center**. It is usually selectable from the bar at the buttom of the screen or by invoking the command "**drakconf**" from a terminal:



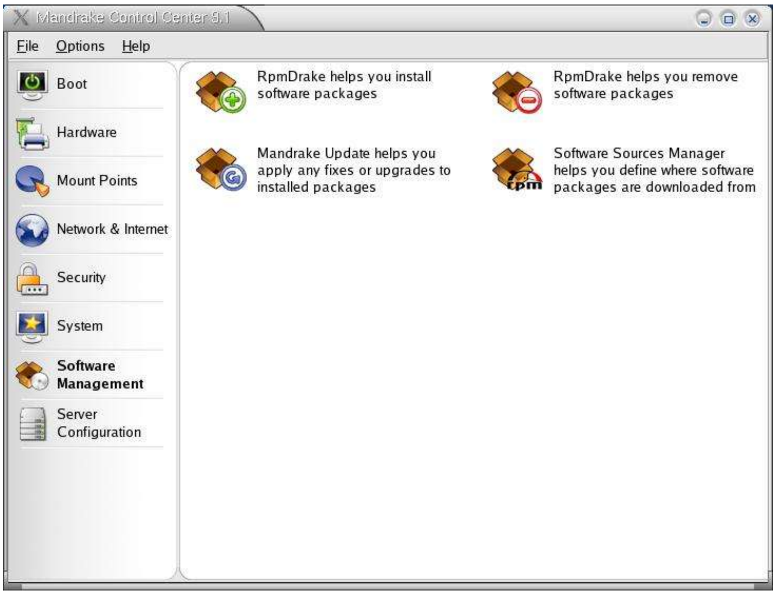
Welcome to the Mandrake Control Center

Mandrake Control Center is Mandrake Linux's main configuration tool. It enables the system administrator to configure the hardware and services used for all users.

The tools accessed through the Mandrake Control Center greatly simplify the use of the system, notably by avoiding the use of the evil command line.

Illustration 22 The Mandrake Control Center again.

You can use **Software Management** entry from this screen to install the **DNS** Server (if not already installed that is).You will get a screen as below where you click on the **RpmDrake** icon to install the software we need:



RpmDrake helps you install software packages

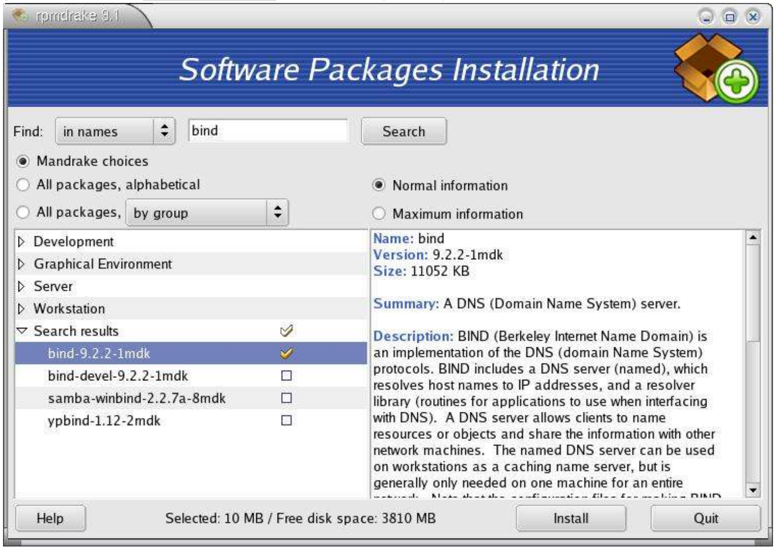
RpmDrake helps you remove software packages

Mandrake Update helps you apply any fixes or upgrades to installed packages

Software Sources Manager helps you define where software packages are downloaded from

Illustration 23 Choose to install software here

As with the **DHCP** server earlier, we search for **BIND** (not **DNS**) in this case, which (after you have pressed the **Search** button) gives you these results:



Find: in names bind

● Mandrake choices
○ All packages, alphabetical
○ All packages, by group

● Normal information
○ Maximum information

Search results ✓
bind-9.2.2-1mdk ✓
bind-devel-9.2.2-1mdk □
samba-winbind-2.2.7a-8mdk □
ypbind-1.12-2mdk □

Name: bind
Version: 9.2.2-1mdk
Size: 11052 KB

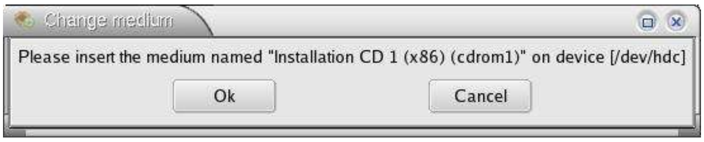
Summary: A DNS (Domain Name System) server.

Description: BIND (Berkeley Internet Name Domain) is an implementation of the DNS (domain Name System) protocols. BIND includes a DNS server (named), which resolves host names to IP addresses, and a resolver library (routines for applications to use when interfacing with DNS). A DNS server allows clients to name resources or objects and share the information with other network machines. The named DNS server can be used on workstations as a caching name server, but is generally only needed on one machine for an entire

Selected: 10 MB / Free disk space: 3810 MB

Illustration 24 The search results for the bind search string

As you can see, we already selected the only option required to install DNS services on our system: `bind-9.2.2-1mdk`. Click the **Install** button and you will be prompted to insert **Installation CD 1** from the Download set and again follow the instruction on screen to proceed.



Please insert the medium named "Installation CD 1 (x86) (cdrom1)" on device [/dev/hdc]

Illustration 25 Load Installation CD 1 for BIND

7.3 Where did everything go?

The **BIND9** package provided with the Mandrake distribution follows the standard where file locations are concerned. The basic configuration file: `named.conf` is by default located in the directory `/etc` and the *zone* files are by default under the `/var/named` directory. This `/var/named` directory is specified in the `named.conf` file as we will see later.

7.4 The main configuration file `named.conf`

This `named.conf` file is the main configuration file for **BIND**, The file lists, apart for some global settings, the *zones* for which this **DNS** server has some information to spare. Below you will find the `named.conf` file we used in our lab setup. The files mentioned for the *zones* have a “syntax” that is similar to the one found in the O' Reilly handbook "**DNS and BIND 4th Edition**" i.e. we use `db.zone-name` as the file name for the *zones* used. This is not mandatory but it helps to stick to some sort of standard:

[/etc/named.conf]

```
options {
        directory "/var/named";
        pid-file "/var/named/named.pid";
        };

# name to address
zone "domain.org"{
        type master;
        file "db.private";
        allow-update { 192.168.1.1; };
        };

# address to name
zone "1.168.192.in-addr.arpa"{
        type master;
        file "db.192.168.1";
        allow-update { 192.168.1.1; };
};

# localhost to 127.0.0.1
zone "localhost"{
        type master;
        file "db.localhost";
        };# 127.0.0.1 to localhost
        };

# 127.0.0.1 to localhost
zone "0.0.127.in-addr.arpa"{
        type master;
        file "db.127.0.0";
        };

# dummy zones
zone "0.in-addr.arpa"{
        type master;
        file "db.dummy";
        };

zone "255.in-addr.arpa"{
        type master;
        file "db.dummy";
        };

zone "." {
        type hint;
        file "named.root";        };
```

A little explanation should go with this file (I guess). The first lines define some global options, one of them is defining the "*root*" of the *named* configuration files. This option indicates that the *zone* files can be found under the directory `/var/named`, and as a result you should find the *zones* specified in this file under the `/var/named` directory. For example there is a *zone* defined with `db.localhost` as the name. There should be a file called `/var/named/db.localhost` for that matter. Also note that for the `db.private` and `db.192.168.1` files a statement was added to allow updates from our **DHCP** server. As you can see, `named.conf` is only just the start, it defines the *zones* and these can be a real pain to setup. The examples can be used as a starting point for your specific setup. We will go (a little) deeper into these *zone* files next.

7.5 The zone files

Zone files are quite important to **DNS** as they will specify to **DNS** what to do with request received from clients. Some special *zone* files should always be part of your **DNS** configuration and you can use the ones listed here in the example, other are specific to your specific situation and must be tinkered with. Some things just can't be helped... Let's have a look at the *zone* files we created in our lab setup. Since **DNS** is used to resolve names to valid IP addresses, there must be some information on how to achieve this. This is done via the *zone* files. A *zone* can have a function to map IP addresses to names while another *zone* can have the opposite function (a *reverse zone* as it is called). So one would expect to find for each *zone* two files that handle requests for that *zone*.

7.5.1 Some specific "overhead" *zones*

As mentioned, some *zones* should always be part of your **DNS** server configuration. There are sort of standard overhead but you need them to take care of some standard stuff. Amongst these are *zones* that deal with the "*localhost*", the network name, the broadcast address and of course the big bad world outside (the "root" *zone*).

7.5.1.1 The “localhost” and the (reverse lookup) ”0.0.127.in-addr-arpa" *zone*

These *zones* for instance, will cover the *loopback* name resolving. In the example configuration we used the *zone* was named "`db.localhost`" while the *reverse lookup* was handled by a files that was named "`db.127.0.0`". These files are a must to allow lookups to the *loopback* or *localhost* address and vice versa. No DNS should be without these two. First as a little refresh, the references to these files in `/etc/named.conf`:

[/etc/named.conf (part)**]**

```
# localhost to 127.0.0.1
zone "localhost"{
        type master;
        file "db.localhost";
        };# 127.0.0.1 to localhost
        };

zone "0.0.127.in-addr.arpa"{
        type master;
        file "db.127.0.0";
        };
```

The contents of the `db.localhost` and `db.127.0.0` files are listed here. We start with the `db.localhost` file:

[/var/named/db.localhost]

```
$TTL 259200
@ IN SOA pookie.domain.org. hostmaster.domain.org. (
2003071002 ;serial YYYYMMDDxx set by DNS admin (you!!)
86400      ;refresh once a day same 1d
3600       ;retry once per hour same as 1h
604800     ;expire after one week same as 1w
259200)    ;minumum lifespan three days same as 3d
;
      IN NS pookie.domain.org.
localhost.    IN A 127.0.0.1
```

Here is the *reverse lookup file* for "localhost" which is `db.127.0.0`:

[/var/named/db.127.0.0]

```
$TTL 259200
@      IN      SOA     pookie.domain.org. hostmaster.domain.org. (
2003071002 ;serial YYYYMMDDxx set by DNS admin (you!!)
86400      ;refresh once a day same 1d
3600       ;retry once per hour same as 1h
604800     ;expire after one week same as 1w
259200)    ;minumum lifespan three days same as 3d
;
0.0.127.in-addr.arpa. IN NS  pookie.domain.org.
1.0.0.127.in-addr.arpa.      IN PTR         localhost.
```

7.5.1.2 The “root” zone

This is also a very important *zone* if your DNS takes care of connections to the outside world. This *zone* which out of your control will help you to resolve named like www.Invensys.com to a valid IP adress. These “root” *zones* are served by the internet “root” name servers. A list of these servers is available on the internet and should be updated regularly. You can get the latest version through the location: "`ftp.internic.net`" under the directory `/domain`, the file to look for is "`named.root`". The file we used is of course located in the `/var/named` and (in our example setup) has a filename `named.root`. It is referenced in the `/etc/named.conf` file:

[/etc/named.conf (part)**]**

```
zone "." {
        type hint;
        file "named.root";        };
```

Note that this *zone* is of type "*hint*" as we do not know if **DNS** will find the answer you are looking for in that zone, it is indeed no more than just a hint for **DNS** to have a look there.
The file "`named.root`" contains these hints (the "root name servers") and it is listed here:

[/var/named/named.root]

```
;       This file holds the information on root name servers needed to
;       initialize cache of Internet domain name servers
;       (e.g. reference this file in the "cache  .  <file>"
;       configuration file of BIND domain name servers).
;
;       This file is made available by InterNIC
;       under anonymous FTP as
;           file                /domain/named.root
;           on server           FTP.INTERNIC.NET
;
;       last update:    Nov 5, 2002
;       related version of root zone:   2002110501
;
;
; formerly NS.INTERNIC.NET
```

```
;
.                        3600000  IN  NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.      3600000      A     198.41.0.4
;
; formerly NS1.ISI.EDU
;
.                        3600000      NS    B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.      3600000      A     128.9.0.107
;
; formerly C.PSI.NET
;
.                        3600000      NS    C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET.      3600000      A     192.33.4.12
;
; formerly TERP.UMD.EDU
;
.                        3600000      NS    D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET.      3600000      A     128.8.10.90
;
; formerly NS.NASA.GOV
;
.                        3600000      NS    E.ROOT-SERVERS.NET.
E.ROOT-SERVERS.NET.      3600000      A     192.203.230.10
;
; formerly NS.ISC.ORG
;
.                        3600000      NS    F.ROOT-SERVERS.NET.
F.ROOT-SERVERS.NET.      3600000      A     192.5.5.241
;
; formerly NS.NIC.DDN.MIL
;
.                        3600000      NS    G.ROOT-SERVERS.NET.
G.ROOT-SERVERS.NET.      3600000      A     192.112.36.4
;
; formerly AOS.ARL.ARMY.MIL
;
.                        3600000      NS    H.ROOT-SERVERS.NET.
H.ROOT-SERVERS.NET.      3600000      A     128.63.2.53
;
; formerly NIC.NORDU.NET
;
.                        3600000      NS    I.ROOT-SERVERS.NET.
I.ROOT-SERVERS.NET.      3600000      A     192.36.148.17
;
; operated by VeriSign, Inc.
;
.                        3600000      NS    J.ROOT-SERVERS.NET.
J.ROOT-SERVERS.NET.      3600000      A     192.58.128.30
;
; housed in LINX, operated by RIPE NCC
;
.                        3600000      NS    K.ROOT-SERVERS.NET.
K.ROOT-SERVERS.NET.      3600000      A     193.0.14.129
;
; operated by IANA
;
.                        3600000      NS    L.ROOT-SERVERS.NET.
L.ROOT-SERVERS.NET.      3600000      A     198.32.64.12
;
; housed in Japan, operated by WIDE
;
.                        3600000      NS    M.ROOT-SERVERS.NET.
M.ROOT-SERVERS.NET.      3600000      A     202.12.27.33
; End of File
```

7.5.1.3 The "dummy" zone

This *zone* is required to handle request to the network or broadcast address. As we are used to, it is referenced in the `/etc/named.conf` file:

```
[/etc/named.conf (part)]
# dummy zones
zone "0.in-addr.arpa"{
        type master;
        file "db.dummy";
        };

zone "255.in-addr.arpa"{
        type master;
        file "db.dummy";
        };
```

It does not contain anything usefull and therefore all requests will result in a simple "not found".
The content of `/var/named/db.dummy`:

```
[/var/named/db.dummy]
$TTL 259200
@ IN SOA pookie.domain.org. hostmaster.domain.org. (
2003071002 ;serial YYYYMMDDxx set by DNS admin (you!!)
86400      ;refresh once a day same 1d
3600       ;retry once per hour same as 1h
604800     ;exipire after one week same as 1w
259200)    ;minumum lifespan three days same as 3d
;
        NS pookie.domain.org.
        PTR localhost.
```

7.5.2 The zones that matter :-)

After dealing with the overhead stuff, we can get down to the real thing. We need some *zones* for the work we are trying to do here. These *zones* are for dealing with our domain **domain.org** and should handle requests for that *zone*. One zone we have is titled `domain.org` and, as we are used to by now,(?) there is also a reverse lookup file and that is zone `1.168.192.in-addr-arpa`. These *zones* we need to setup too and since we did that already for you let's have a look:
First see what the files are in `/etc/named/conf`:

```
[/etc/named.conf (part)]
zone "domain.org"{
        type master;
        file "db.private";
        allow-update { 192.168.1.1; };
        };

# address to name
zone "1.168.192.in-addr.arpa"{
        type master;
        file "db.192.168.1";
        allow-update { 192.168.1.1; };
};
```

This **DNS** Server considers itself a *master* for these *zones* and the names for the files where **DNS** can find info about this *zone* are `db.private`. and `db.192.168.1`. Ergo, these files must exsist in the directory `/var/named` (remember that this was declared the "root" for *named* in the `named.conf` file. Note the lines that read "`allow-update { 192.168.1.1; } ;`" in the `/etc/named.conf` file for these two *zones* which are the lines that will make **DNS** allow updates of these zones from your **DHCP** server.
The "`db.private`" zone file we used:

```
[/var/named/db.private]
$TTL 259200
@      IN     SOA    pookie.domain.org. hostmaster.domain.org. (
2003071003 ;serial YYYYMMDDxx set by DNS admin (you!!)
86400      ;refresh once a day same as 1d
3600       ;retry once per hour same as 1h
604800     ;exipire after one week same as 1w
259200)    ;minumum lifespan three days same as 3d
;
              NS      pookie.domain.org.
;
pookie IN     A       192.168.1.1
AW51B  IN     A       192.168.1.2
; aliases
mozart IN     CNAME   pookie
print  IN     CNAME   pookie
```

As we can see, the names "**pookie**" and "**AW51B**" both have fixed IP addresses in this configuration. Please note that there is no mentioning of anything related to the X-Terminals. Of course we also need the *reverse lookup* file for this network and that would be the file "`db.192.168.1`" which is exactly what we used:

[/var/named/db.192.168.1]

```
$TTL 259200   ; 3 days
@      IN SOA  pookie.domain.org. hostmaster.domain.org. (
                             2003071007 ; serial
                             86400      ; refresh (1 day)
                             3600       ; retry (1 hour)
                             604800     ; expire (1 week)
                             259200     ; minimum (3 days)
                             )
                      NS      pookie.domain.org.
1                        PTR     localhost.
2                        PTR     AW51B.
```

7.6 The initial zone files

Before we start the **DNS** services, we take a look at the `/var/named` directory to see what we have so far. We should at least have these files:

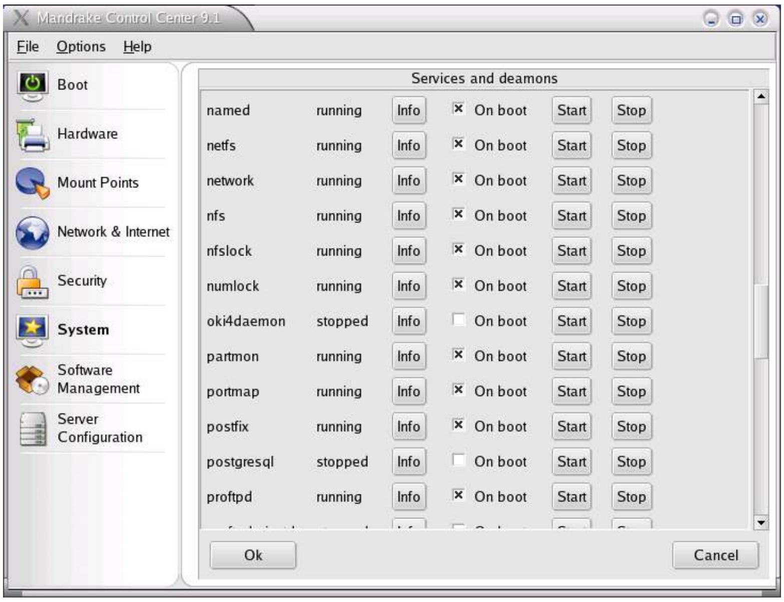
[The contents of the /var/named directory before running DNS/DHCP]

```
db.127.0.0
db.192.168.1
db.dummy
db.localhost
db.private
named.root
```

There is a reason why we do this of course. These files are supposed to change if everything goes well and we did the configuration work correctly. Remember we need DNS to dynamic? Well at this point no dynamics are required (nothing is running) but as soon as a client will request an IP address, the DHCP Server will attempt to update the DNS server of this and the *zone* files which are open for updates will reflect this change.

7.6.1 Starting DNS (named)

When all configuration work is done, we can try to start the service. Again similar to the **DHCP** service, **DNS** can be started from the **Mandrake Control Center**. Select the **System** Icon and after selecting the **DrakXServices** icon, scroll down until you get the **named** program. The same as with **dhcpd** applies: run at boot and try to start the service now.



named	running	Info	☒ On boot	Start	Stop
netfs	running	Info	☒ On boot	Start	Stop
network	running	Info	☒ On boot	Start	Stop
nfs	running	Info	☒ On boot	Start	Stop
nfslock	running	Info	☒ On boot	Start	Stop
numlock	running	Info	☒ On boot	Start	Stop
oki4daemon	stopped	Info	□ On boot	Start	Stop
partmon	running	Info	☒ On boot	Start	Stop
portmap	running	Info	☒ On boot	Start	Stop
postfix	running	Info	☒ On boot	Start	Stop
postgresql	stopped	Info	□ On boot	Start	Stop
proftpd	running	Info	☒ On boot	Start	Stop

Illustration 26 DrakXServices view in Mandrake Control Center- on top the "named" configuration

7.7 Verifying proper operation

To see whether we where successfull in our setup it is time to check this out. We assume the following situation:

- The **I/A 50 Series** host is up and running
- The **DNS** and **DHCP** servers are both on-line

To see how things are going in may be a nice idea to have a `tail -f` on the `/var/log/messages` file on the Linux machine. It is time to boot up the **Windows XP** X-Terminal PC...

7.7.1 The zone files after running DHCP and DNS and client request

When the **Windows XP** machine is on-line, you should see that the IP address is handed out by the **DHCP** Server and also you should see that **DHCP** is attempting to update the **DNS** files related to the `db.private` and `db.192.168.1` zones in `/var/named` and **DNS** should respond in favor of these attempts. Another change you may observe is some additional files in `/var/named` directory showed up on the Linux box:

[The contents of the /var/named directory after booting the XP machine]

```
db.127.0.0
db.192.168.1
db.192.168.1.jnl
db.dummy
db.localhost
db.private
db.private.jnl
named.pid
named.root
```

As we can see, there are some extra files which are the direct result from the interaction between **DHCP** and **DNS** on this system. The files with the extention `.jnl` are the journal files that are used by **DNS** to maintain the (now dynamic) **DNS** server files. Don't be surprised to see these db.private.jnl and db.192.168.1.jnl files on your system since they are proof of your success. More important: do not just delete them.

7.7.2 The db.private zone after a dynamic update

One of the *zones* we have that will be updated by **DHCP** is the db.private *zone*. Once the **Windows XP** PC is on-line and has received an IP address, there should a change to the db.private *zone* file in addition to the db.private.jnl file. Note that the db.private.jnl file is binary and is not user readable. The journal file is used to maintain periodic updates of the *zone* files in **DNS**.

7.7.3 Messages in /var/log/messages

After starting the DHCP Server, your /var/log/messages file should contain some lines similar to this:

[/var/log/messages snippet]

```
Aug 20 08:46:00 Pookie dhcpd: dhcpd shutdown failed
Aug 20 08:46:00 Pookie dhcpd: Wrote 0 deleted host decls to leases file.
Aug 20 08:46:00 Pookie dhcpd: Wrote 0 new dynamic host decls to leases file.
Aug 20 08:46:00 Pookie dhcpd: Wrote 0 leases to leases file.
Aug 20 08:46:00 Pookie dhcpd:
Aug 20 08:46:00 Pookie dhcpd: No subnet declaration for eth0 (10.31.19.230).
Aug 20 08:46:00 Pookie dhcpd: ** Ignoring requests on eth0.  If this is not what
Aug 20 08:46:00 Pookie dhcpd:    you want, please write a subnet declaration
Aug 20 08:46:00 Pookie dhcpd:    in your dhcpd.conf file for the network segment
Aug 20 08:46:00 Pookie dhcpd:    to which interface eth0 is attached. **
Aug 20 08:46:00 Pookie dhcpd:
Aug 20 08:46:00 Pookie dhcpd: dhcpd startup succeeded
```

When you run the **DNS** Server next, some extra lines should appear in this file:

[/var/log/messages snippet]

```
Aug 20 08:48:48 Pookie named: named shutdown failed
Aug 20 08:48:48 Pookie named[8253]: starting BIND 9.2.2 -u named
Aug 20 08:48:48 Pookie named[8253]: using 1 CPU
Aug 20 08:48:48 Pookie named: named startup succeeded
Aug 20 08:48:49 Pookie named[8255]: loading configuration from '/etc/named.conf'
Aug 20 08:48:49 Pookie named[8255]: no IPv6 interfaces found
Aug 20 08:48:49 Pookie named[8255]: listening on IPv4 interface lo, 127.0.0.1#53
Aug 20 08:48:49 Pookie named[8255]: listening on IPv4 interface eth0, 10.31.19.3
Aug 20 08:48:49 Pookie named[8255]: listening on IPv4 interface eth1, 192.168.13
Aug 20 08:48:49 Pookie named[8255]: zone 'domain.org' allows updates by IP addre
Aug 20 08:48:49 Pookie named[8255]: zone '1.168.192.in-addr.arpa' allows updatee
Aug 20 08:48:49 Pookie named[8255]: couldn't add command channel 127.0.0.1#953:d
Aug 20 08:48:49 Pookie named[8255]: zone 0.in-addr.arpa/IN: loaded serial 200302
Aug 20 08:48:49 Pookie named[8255]: zone 0.0.127.in-addr.arpa/IN: loaded serial2
Aug 20 08:48:49 Pookie named[8255]: dns_master_load: db.192.168.1:9: 1.168.192.e
Aug 20 08:48:49 Pookie named[8255]: zone 1.168.192.in-addr.arpa/IN: loading mase
Aug 20 08:48:49 Pookie named[8255]: zone 255.in-addr.arpa/IN: loaded serial 2002
Aug 20 08:48:49 Pookie named[8255]: zone localhost/IN: loaded serial 2003071002
Aug 20 08:48:49 Pookie named[8255]: dns_master_load: db.private:10: domain.org.e
Aug 20 08:48:49 Pookie named[8255]: zone domain.org/IN: loading master file db.e
Aug 20 08:48:49 Pookie named[8255]: running
```

So far so good... Now boot the Windows XP PC and see what happens: When **Windows XP** boots it will request an IP adress and some work needs to be done by **DHCP** and **DNS**:

[/var/log/messages when XP boots]

```
Aug 20 11:58:05 Pookie dhcpd: DHCPDISCOVER from 00:60:b0:c4:bb:86 (AWRD03) via
eth1
Aug 20 11:58:06 Pookie dhcpd: DHCPOFFER on 192.168.1.25 to 00:60:b0:c4:bb:86
(AWRD03) via eth1
Aug 20 11:58:06 Pookie named[1227]: client 192.168.1.1#32781: updating zone
'domain.org/IN': adding an RR
```

```
Aug 20 11:58:06 Pookie named[1227]: client 192.168.1.1#32781: updating zone
'domain.org/IN': adding an RR
Aug 20 11:58:06 Pookie dhcpd: if IN A AWRD03.domain.org domain doesn't exist add
150 IN A AWRD03.domain.org 192.168.1.25 add 150 IN TXT AWRD03.domain.org
"3193fe5d3a8fa5d6a29ab668812c969fae": success.
Aug 20 11:58:06 Pookie named[1227]: client 192.168.1.1#32781: updating zone
'1.168.192.in-addr.arpa/IN': deleting an rrset
Aug 20 11:58:06 Pookie named[1227]: client 192.168.1.1#32781: updating zone
'1.168.192.in-addr.arpa/IN': adding an RR
Aug 20 11:58:06 Pookie dhcpd: delete IN PTR 25.1.168.192.in-addr.arpa. add 150 IN
PTR 25.1.168.192.in-addr.arpa. AWRD03.domain.org: success.
Aug 20 11:58:06 Pookie dhcpd: DHCPREQUEST for 192.168.1.25 (192.168.1.1) from
00:60:b0:c4:bb:86 (AWRD03) via eth1
Aug 20 11:58:06 Pookie dhcpd: DHCPACK on 192.168.1.25 to 00:60:b0:c4:bb:86
(AWRD03) via eth1
```

You can (after some time that is) have a look at the `db.private` and `db.192.168.1` files. Note that these files lag behind the real world because **DNS** does not update these files immediately after every change. (this would cause a lot of stress on the disk when a lot of clients are requesting IP addresses). **DNS** keeps track of the IP addresse/name mapping in cache and updates the *zone* file every hour or so. So, after some time the files may look something like this:

Before dynamic updates:

/var/named/db.private before dynamic update

```
$TTL 259200     ; 3 days
domain.org              IN SOA  pookie.domain.org. hostmaster.domain.org. (
                                2003071010 ; serial
                                86400      ; refresh (1 day)
                                3600       ; retry (1 hour)
                                604800     ; expire (1 week)
                                259200     ; minimum (3 days)
                                )
;
                        NS      pookie.domain.org.
;
pookie          IN      A       192.168.1.1
AW51B           IN      A       192.168.1.2
; aliases
mozart          IN      CNAME   pookie
print           IN      CNAME   pookie
```

and after:

/var/named.db.private after dynamic update:

```
domain.org              IN SOA  pookie.domain.org. hostmaster.domain.org. (
                                2003071047 ; serial
                                86400      ; refresh (1 day)
                                3600       ; retry (1 hour)
                                604800     ; expire (1 week)
                                259200     ; minimum (3 days)
                                )
                        NS      pookie.domain.org.
$ORIGIN domain.org.
AW51B                   A       192.168.1.2
$TTL 30 ; 30 seconds
AWRD03                  A       192.168.1.25
                        TXT     "3193fe5d3a8fa5d6a29ab668812c969fae"
$TTL 259200     ; 3 days
mozart                  CNAME   pookie
pookie                  A       192.168.1.1
print                   CNAME   pookie
```

Similar updates are expected for the “reverse lookup” files

That's it for now

RdG/RD