# Coyote Linux 1.4
# User Documentation

Revision 1.0-4.8.03 (draft)

Author: Chris Stoll chris.stoll@luk-us.com

Date: 4/8/2003

**TABLE OF CONTENTS**

# Introduction

Coyote Linux is a single floppy distribution of Linux that is designed for the purpose of sharing an Internet connection. In addition to connection sharing, it also provides firewall services to help protect the internal network. The goal of the Coyote project is to make it as quick and easy as possible to share an Internet connection. The floppy can be created using either a Microsoft Windows "wizard", or by using a set of Linux shell scripts. In addition to being designed to have very low hardware requirements, the floppy release of Coyote Linux is able to provide the performance and uptime that is expected from any Linux based system.

## *Coyote Linux Features*

| | Linux Disk Creator | Windows Disk Creator |
|---|---|---|
| **DHCP Connection** | ✓ | ✓ |
| **Static IP Connection** | ✓ | ✓ |
| **PPP Dialup** | ✓ | ✓ |
| **ISDN Support** | ✓ | ✓ |
| **PPPoE Connection** | ✓ | ✓ |
| **DHCP Server** | ✓ | ✓ |
| **IP Chains Firewalling** | ✓ | ✓ |
| **IP Auto-forwarding** | ✓ | ✓ |
| **Web Based Admin** | ✓ | ✓ |
| **PPTP Client Support**[1] | ✓ | ✓ |

[1] This version of Coyote supports the use of PPTP based VPN software on the local area network, but does not contain an internal PPTP client.

## *Licensing*

Coyote Linux was created by Joshua Jackson and is distributed under the GNU Public License (GPL). The Windows Wizard, also created by Joshua Jackson, is open-source, freeware but is not covered under the GPL. If you intend to use the source code for the Wizard in one of your own products, you need to obtain permission before doing so.

For technical support of items not covered in this manual, try the Coyote Linux forum.

## *Credits*

Portions of this documentation were derived from:
Original Coyote Linux documentation and the Coyote Linux Forum maintained by Joshua Jackson.
The BusyBox (http://www.busybox.net/) website which is maintained by Erik Andersen.
Linux IPCHAINS-HOWTO (http://www.netfilter.org/ipchains/HOWTO.html) by Rusty Russell

# Getting started with Coyote Linux

## *System Requirements*

Coyote Linux is designed to be run on Intel x86 based systems with the following minimum system requirements:

386SX or better[1]
12Mb RAM
Two network cards[2]
1.44Mb Floppy disk drive
MDA (Monochrome Display Adapter)

Optional components:

CDROM drive
Compact Flash drive (this requires hard drive support, a pending feature)

[1] A math coprocessor is required if you are using the Windows Disk Creator. Math-coprocessor emulation is not built into the Windows Disk creator version because it requires considerably more disk space.
[2] Appendix A gives a listing of supported network adapters. SYSLINUX, the Linux boot loader can have problems with some network cards containing a boot ROM. If you are having trouble with a card that has a boot ROM try removing the ROM or the card altogether.

## *Installing Coyote Linux*

To create a Coyote Linux floppy, there are two options available. You can chose from either a Microsoft Windows Wizard or a Linux shell script to create the floppy with your desired options. While the Wizard will be a much easier option for users that are familiar with Microsoft Windows, its functionality is somewhat limited in comparison to the Linux floppy creator. The difference in the products are mainly due to limitations in Microsoft Windows itself.

To start the installation process you will need to order the Coyote Linux CD from Vortech Consulting. (http://www.vortech.net/) Or, alternatively, you can download the Windows Disk Creator or the Linux Floppy Creator Scripts from the Coyote Linux website. (http://www.coyotelinux.com/)

## Using the Windows Disk Creator

Unzip the Windows Disk Creator to the desired location and then execute Coyote.exe.

Once the application launches you will be greeted with an information screen, click next to begin the floppy creation process.

The Windows disk creator is very easy to use, but I will briefly go through the steps for completeness sake.

### Step 1

Here you will be asked for your LAN configuration. You should not need to make any changes here unless you need more than 253 internal IP numbers or wish to change the internal IP address of the router. If you are not familiar with networking it is best to leave these settings as they are.

If you do decide to change these settings be sure to use one of the three private, non-routable ranges:

| | **Start** | **Ending IP** | **Networks** | **Devices** |
|---|---|---|---|---|
| **Class A** | 10.0.0.1 | 10.255.255.254 | 1 | 16,777,214 |
| **Class B** | 172.16.0.1 | 172.31.255.254 | 16 | 65,534 |
| **Class C** | 192.168.0.1 | 192.168.255.254 | 255 | 254 |

### Step2

Now you will be asked to enter your password. It is required because this will be needed to access your router via ssh and the web admin page. Choose a strong password and remember that your password is case sensitive.

### Step 3

Here you are given the option to send you syslog messages to a remote machine. This field is not required.

### Step 4

In step three you must specify which method you use to connect to your ISP. Once you choose the connection type further options will be presented to you. For the Static IP, PPPoE DSL, and PPP modem dialup options you will need to know the name of your name servers and the login domain. This information should have been provided by your ISP. For PPPoE and PPP modem dialup you will also have to provide your username and password. For the static IP configuration you will need to know your IP address, subnet mask, and default gateway. This is also provided by your ISP.

### Step 5

Here you have the option to enable the DHCP server for your internal network. To accomplish this simply check the box and then specify how many internal host you need to have available. You may want to take the default number even if you don't have 189 hosts on your internal network, this will give you room to grow and will not affect your performance.

### Step 6

Now you must enter the type of network cards you plan on using. Only a few ISA cards may require additional parameters, for most of the cards leave the IO address and IRQ blank.

Note: It is recommended to use two different makes of network cards, this make troubleshooting problems a lot easier.

### Step 7

Finally, insert a blank disk in your drive and click 'Create Disk'.

You are now ready to boot your Coyote Linux machine.

## Using the Linux Floppy Creator Scripts

Unzip and untar the Linux Floppy Creator Scripts to the desired location. Su to root and then execute makefloppy.sh.

The Linux Floppy Creator Scripts work in just about the same fashion as the Windows Disk Creator with a few exceptions. The Linux Floppy Creator Scripts will ask you which disk size you would like to use, 1.68Mb is generally the best choice. Also, you will be given the option to use a kernel that has math co-processor emulation built into it. This would allow you to use a math co-processor-less machine.

### Step 1

If you are attempting to create the Coyote Linux disk as a non-super user you will be presented with the following warning:

```
Coyote floppy builder script v2.6

You should be logged in as root to create the Coyote floppy disk. Non-root users
can experience permission problems under some distributions when attempting to
access the floppy drive.  To cancel, press CTRL-C... to proceed, press enter
```

Next you will be asked which size of disk you wish to create:

```
Please choose the desired capacity for the created floppy:
```

```
1) 1.44Mb (Safest and most reliable but may lack space needed for
           some options)
2) 1.68Mb (Good reliability with extra space) - recommended
3) 1.72Mb (Most space but may not work on all systems or with all
           diskettes)

Enter selection: 2
```

### Step 2

Next, you will be asked which type of processor Coyote Linux will be using:

```
Please select the processor type in the destination Coyote Linux
system:

1) 386sx, 386dx, 486sx (No math co-processor)
2) 486dx or better (has a math co-processor)

Enter Selection: 2
```

### Step 3

Now you will be asked which type of connection you have to the internet:

```
Please select the type of Internet connection that your system uses.

1) Standard Ethernet Connection
2) PPP over Ethernet Connection
3) PPP Dialup Connection

Enter Selection: 2

Configuring system for PPP over Ethernet.
```

### Step 4

This step gives you the opportunity to change the network configuration. For most small office or home office networks these settings will not need to be changed.

If you do decide to change these settings be sure to use one of the three private, non-routable ranges:

| | Start | Ending IP | Networks | Devices |
|---|---|---|---|---|
| **Class A** | 10.0.0.1 | 10.255.255.254 | 1 | 16,777,214 |
| **Class B** | 172.16.0.1 | 172.31.255.254 | 16 | 65,534 |
| **Class C** | 192.168.0.1 | 192.168.255.254 | 255 | 254 |

Here is what you will be shown:

```
By default, Coyote uses the following settings for the local network
interface:

IP Address: 192.168.0.1
Netmask:    255.255.255.0
Broadcast:  192.168.0.255
Network:    192.168.0.0

Would you like to change these settings? [Y/N]: N
```

**Step 5**

What you see next depends on the type of connection you have with your ISP. Here is the dialog for a PPPoE type connection:

```
Enter PPPoE username: isp_assigned_name

Enter PPPoE password: isp_password

Enter the domain name for your area: isp_domain

Enter your primary DNS server IP: 1.2.3.4

Enter your secondary DNS server IP: 1.2.3.5

You can either
- keep up the connection permanently [1]
- or connect automatically on outbound traffic
  and close the connection after a period of
  inactivity that you can define     [2]

Which option do you want [1/2]: 1
```

**Step 6**

No user input is required here, The script will automatically create your ssh host key:

```
Generating Coyote host key...

Initializing random number generator...
Generating p:  ...........++ (distance 166)
Generating q:  .++ (distance 36)
Computing the keys...
Testing the keys...
Key generation complete.
Your identification has been saved in pkgsrc/etc/etc/ssh/ssh_host_key.
Your public key is:
1024 35
26732653763097438926884237505280662239255193915256222444646637699784912582828666205
46726540194517581695699009441488394294440523184803543182558353138447120879026106987
```

```
11759627900716902865658166600707946038003730935506898613396143791000924495275524250
402564196661433290019073948482929782732244234839048487356145 root@workstation-a
Your public key has been saved in pkgsrc/etc/etc/ssh/ssh_host_key.pub
```

### Step 7

Here you have the option to enable the DHCP server for your internal network (if you changed the network settings in step for you will have to also enter your dhcp scope here):

```
Do you want to enable the coyote DHCP server [y/n]: y
```

### Step 8

Next, we need to enter the type of network cards we will be using. You can look in Appendix A for your network card type to see which heading or module it falls under.

```
You now need to specify the module name and parameters for your network cards.

If you are using PCI or EISA cards, leave the IO and IRQ lines blank.

Enter the module name for you local network card: 3c509
Enter IO address (Leave blank for PCI cards):
Enter IRQ (Leave blank for PCI cards):

Enter the module name for your Internet network card: eepro100
Enter IO address (Leave blank for PCI cards):
Enter IRQ (Leave blank for PCI cards):
Checking module deps for (3c509,eepro100)...
Module 3c509 dep =
Module eepro100 dep =
Copying module: drivers/3c509.o
Copying module: drivers/eepro100.o
Building package: pppoe
Building package: etc
Building package: local
Building package: modules
Building package: root
Building package: dhcpd
Building package: webadmin
```

### Step 9

Insert your floppy disk, hit return, and sit back and wait for your disk to be created.

# Using Coyote Linux

## *System Boot-up*

During boot up, Coyote Linux will display various messages about what it is doing. If your firewall is not performing as expected, be sure to take note of any error messages that are displayed during the boot-up process. If you miss a message you can scroll the screen by using **SHIFT-PGUP** and **SHIFT-PGDN**. Even after the messages are no longer in the screen buffer you can see them again by typing `dmesg` at the Coyote command prompt.

## *System Shutdown*

Unlike most Linux systems, you do not have to prepare a Coyote Linux for shutdown. This is because the root file system is a merely a RAM disk, this means that the root file system is just a portion of the systems physical memory. Further, the boot floppy is unmounted after the boot is complete. Therefore there is no physical file system to be damaged by an abrupt power outage.

So, technically, you can just power down Coyote Linux (like you would have done in the good-old DOS days) using the power switch. But, it is good practice to use the `halt` and `reboot` commands available from the command line. Or you can of course use the same commands from either the console based configuration window or off of the webadmin page.

## *Logging into Coyote Linux*

Once the firewall has booted, on your console you will be given a login prompt that will appear as follows:

```
coyote login:
```

From here you can log in using the username "root". If you used the Windows Disk Creator use the password that you entered in step 2. You will now be presented with the Coyote Linux Gateway – Configuration Menu. (See the section *Logging in Via SSH* below, logging into the console acts exactly the same as logging in via a ssh connection.)

The local logon process logs you into virtual terminal 1, Coyote Linux has 2 other virtual terminals available for you to log into locally. You can change your virtual terminal by pressing **ALT-F2** for virtual terminal 2, **ALT-F3** for virtual terminal 3, and **ALT-F1** to change back to virtual terminal 1. Coyote Linux also displays system kernel messages on virtual terminal 4, to access it press **ALT-F4**. You can also 'scroll' through your terminals by using **ALT-LEFT_ARROW** or **ALT-RIGHT_ARROW**. For these functions you need to use the left **ALT** key. One more trick, **ALT-PrtScr** toggles between the current terminal and the kernel message window, either **ALT** key will work for this particular function.

It is not really necessary to login to Coyote Linux at all, all the standard functionality will work once the machine is booted without user intervention.

It is also possible to log into your Coyote Linux machine remotely at this point. There are two methods available for doing this: the webadmin interface and via ssh.

### *Logging in Via Webadmin*

The webadmin is still in development and some features may be broken or incomplete.

If you took the default setup you can go to this address to log into your router: http://192.168.0.1:8180/

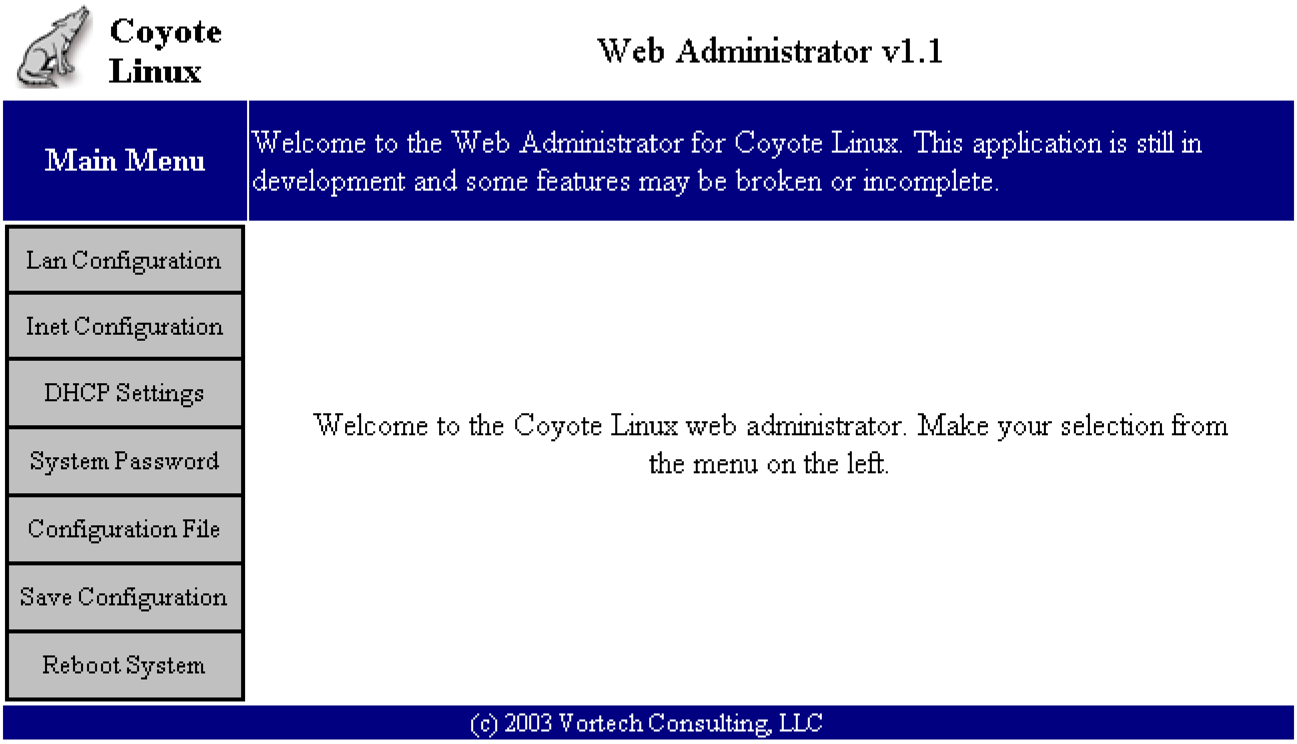
Notice that the port is 8180, not the usual port 80.

Webadmin logon is only available on the LAN side of the router.

You will be required to enter you username (root) and your password. If you used the Windows Disk Creator use the password that you entered in step 2 of the disk creation process to log in.

If you are not asked your password you should log in locally and change your system password via the main configuration menu. This will update all of your logon passwords.

Once you log in click on "Coyote Linux Web Administrator v1.1" and you will be given a screen similar to the one below.



Welcome to the Web Administrator for Coyote Linux. This application is still in development and some features may be broken or incomplete.

Welcome to the Coyote Linux web administrator. Make your selection from the menu on the left.

(c) 2003 Vortech Consulting, LLC

### LAN Configuration

Here you can change your basic network settings: IP Address, Netmask, Network, and Broadcast.

### Inet Configuration

This page allows you to change how Coyote Linux connects to your ISP or the internet.

Currently you can only choose DHCP or static IP configuration.

### DHCP Settings

This function is not yet implemented.

### System Password

This should be self-explanatory.

### Configuration File

This simply allows you to edit your Coyote configuration file, /etc/coyote/coyote.conf.

### Save Configuration

This writes your settings back to the floppy. To use this you should have the floppy in the drive and non-write protected.

This is the only feature that starts acting as soon as you select the menu item, there is no confirmation for this menu item.

### Reboot System

This should be self explanatory. Make sure you floppy is in the drive so that the reboot can succeed.

## *Logging in Via SSH*

Ssh logon is available both on the LAN side of the router and on the external side of the router.

If you used the Windows Disk Creator use the password that you entered in step 2 of the disk creation process to log in. One you are logged in you will be given the exact same configuration menu as if you logged on locally.

Telnet is no longer available, so to access your Coyote Linux box from a windows machine you will need a ssh client, such as PuTTY. (http://www.chiark.greenend.org.uk/~sgtatham/putty/)

If you are not able to log in using ssh you should log in locally and change your system password via the main configuration menu. This will update all of your logon passwords.

## The Configuration Menu

```
                 Coyote Linux Gateway -- Configuration Menu


        1) Edit main configuration file       2) Change system password
        3) Edit firewall script               4) Edit masquerade script (NAT)

        c) Show running configuration         w) Write configuration to floppy
        r) Reboot system

        q) quit
        ------------------------------------------------------------------------
        Selection:
```

From the configuration menu you can access all of the most common administrative items. If you decide to quit this menu you can always return to it by typing `menu` at the command prompt.

### Edit main configuration file

This is the first place to go when you wish to make a change to your network settings. This menu item actually just opens the file /etc/coyote/coyote.conf in the ae editor for you. To save press **`CTRL-s`**, to exit it press **`CTRL-q`**. Below is a table of some common items you will find here.

| Variable Name | Purpose |
|---|---|
| LOCAL_IPADDR | Your internal IP address |
| LOCAL_NETMASK | Your internal network mask |
| LOCAL_BROADCAST | Your internal broadcast address |
| LOCAL_NETWORK | Your internal network |
| USERRM | Use rrlogind daemon flag |
| USEPPPOE | Use PPPoE flag |
| USEDHCP | Use Dynamic Host Configuration Protocol for external IP address flag |
| DNS1 | Primary domain name server name |
| DNS2 | Secondary domain name server name |
| DOMAINNAME | Your domain name |

| | |
|---|---|
| HOSTNAME | Your hostname |
| DCHPHOSTNAME | Your Dynamic Host Configuration Protocol name |
| DHPCSERVER | Use Dynamic Host Configuration Protocol for internal network flag |
| DHCPD_START | Start address of internal Dynamic Host Configuration Protocol range |
| DHCPD_END | End address of internal Dynamic Host Configuration Protocol range |
| GATEWAY | Static default gateway address |

To set your machine's name set both HOSTNAME and DHCPHOSTNAME to the desired name. Do not change DHCPHOSTNAME if your ISP requires this to log on to their network.

### Edit firewall script

This gives you direct access to your firewall startup script. This menu item actually just opens the file /etc/rc.d/rc.firewall in the ae editor for you. To save press `CTRL-s`, to exit it press `CTRL-q`.

### Change system password

This should be self-explanatory.

### Edit firewall script

This is where you should add your custom ipchains entries. You should not make changes to the existing code in the file, simply add your firewall rules after the line that says "ipchains entries go here".

### Edit masquerade script (NAT)

If you need to add port forward or auto forward entries this is where to do so. Additional entries of these type are usually only needed if you intend to run public accessible servers from inside your LAN.

### Show running configuration

This shows the current network configuration.

### Write configuration to floppy

Saves most changes you would make back to the floppy. Make sure the floppy is in the drive and that is write enabled.

### Reboot system

Self-explanatory

## *The MicroEditor e3*

The e3 editor is a simple text editor that is basically the equivalent of Microsoft Windows Notepad. E3 is the editor that is started when you are editing files from the configuration menu. To start e3 from the command line, simply type e3 at the command prompt; or, if you want to edit a particular file, type e3 followed by the name of the file (or the file's path, if the file is not in the working directory). For example, to edit the file filename, type:

```
Coyote# e3 filename
```

Here is a list of commands accepted by e3:

| Command | Purpose |
|---|---|
| `ALT-h` | Help |
| `CTRL-q` | Quit (You'll be asked to save) |
| `CTRL-s` | Save file |
| `CTRL-w` | Write file (Save As) |
| `CTRL-l` | Move to line |
| `CTRL-f` | Find |
| `CTRL-r` | Replace |
| `CTRL-g` | Repeat last Find or Replace |
| `CTRL-a` | Select all |
| `CTRL-x` | Cut |
| `CTRL-c` | Copy |
| `CTRL-v` | Paste |
| `CTRL-u` | Undo |
| `CTRL-e` | Set edit mode |
| `CTRL-k` | Calculate |

You can also have e3 emulate some other popular editors. These are the modes that e3 can run in:

| Mode | Command |
|---|---|
| EMACS emulation | e3em |
| NEdit emulation | e3ne |
| Pico emulation | e3pi |
| vi emulation | e3vi |
| Wordstar emulation | e3ws |

People who are used to using vi (etc.) may wish to add a line like this to their /etc/profile:

```
alias vi="e3vi"
```

**NOTE: When editing Coyote configuration files it is absolutely necessary to include a blank line at the end of the file. Failure to do so may result in major problems! Also, make sure you always use the backup function or your changes will be lost on the next boot of the system!**

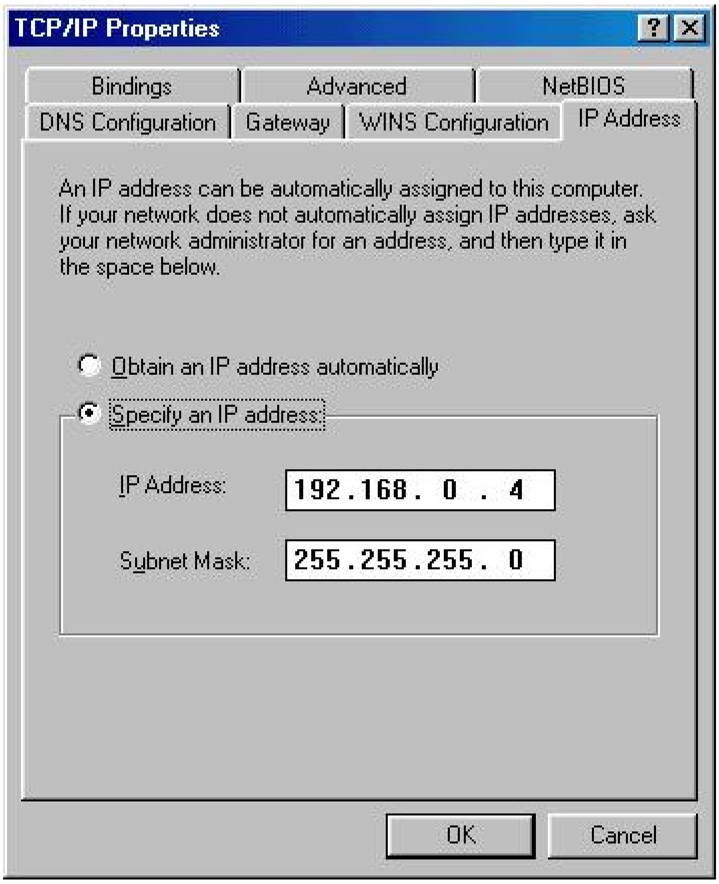
## Configuring Windows Clients

### *Windows 95/98*

The first step is to make sure that you have TCP/IP installed. To check to make sure that it is installed, double click on the network icon from your control panel. You should see a list of your network cards, protocol, and services. If TCP/IP is not listed, click the "add" button, select protocol. From the next dialog, select "Microsoft from the left had column and then TCP/IP from the right hand. Click the OK button and Windows will install TCP/IP for you.

The next step will be determined by your selection of the DHCP server within Coyote. If you selected to use the DHCP server, then your setup of each machine in the network will be simplified.

From the Windows networking dialog (mentioned above), double click on the TCP/IP entry. If you have more than one entry for TCP/IP you probably have the dial-up network adapter installed or Windows has more than 1 network card driver loaded. You will need to select the TCP/IP setting that is bound to your network card that is attached to the network on which you are running the Coyote gateway. The TCP/IP properties box will look like the image below.



TCP/IP Properties

An IP address can be automatically assigned to this computer. If your network does not automatically assign IP addresses, ask your network administrator for an address, and then type it in the space below.

○ Obtain an IP address automatically

● Specify an IP address:

IP Address: 192.168. 0 . 4

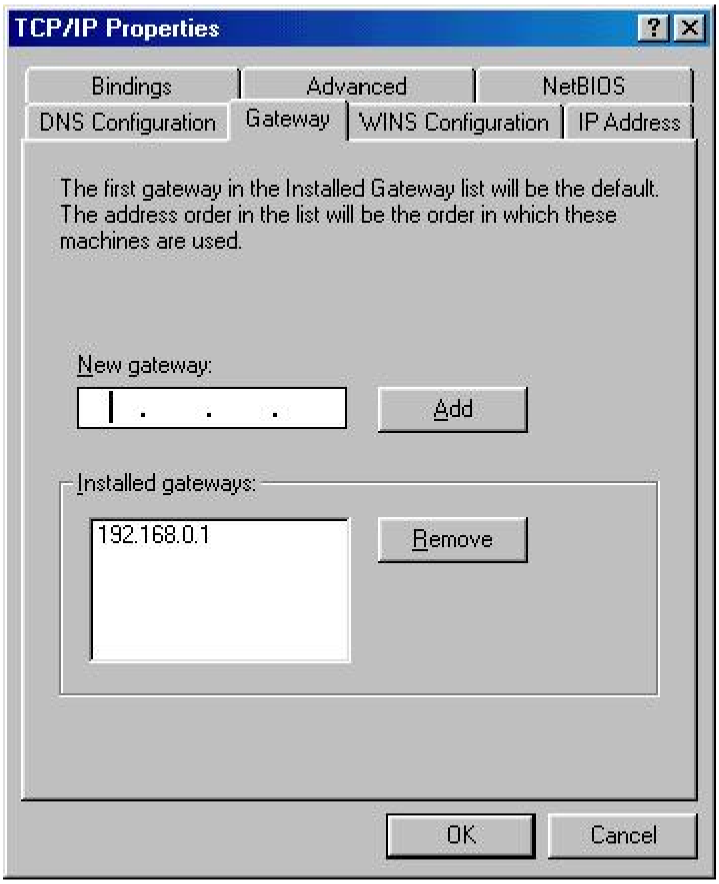
Subnet Mask: 255.255.255. 0

If you selected to use the DHCP server within Coyote, simply check the "obtain an IP address automatically". (The default setup if your Windows computer was directly connected to the cable modem or you have just added TCP/IP to your network protocol list).

If you chose not to use the DHCP server, you will need to specify the address manually. The settings that are used above should closely reflect those that you will need to use.

The only thing that you will need to change from machine to machine will be the "IP Address" entry. These addresses will start at 192.168.0.3 and can be assigned as high as 192.168.0.254. Each machine will need a different IP address. The subnet mask should be the same on all machines (255.255.255.0).

Next, you will need to click on the "gateway" tab. The dialog should look like the following image:



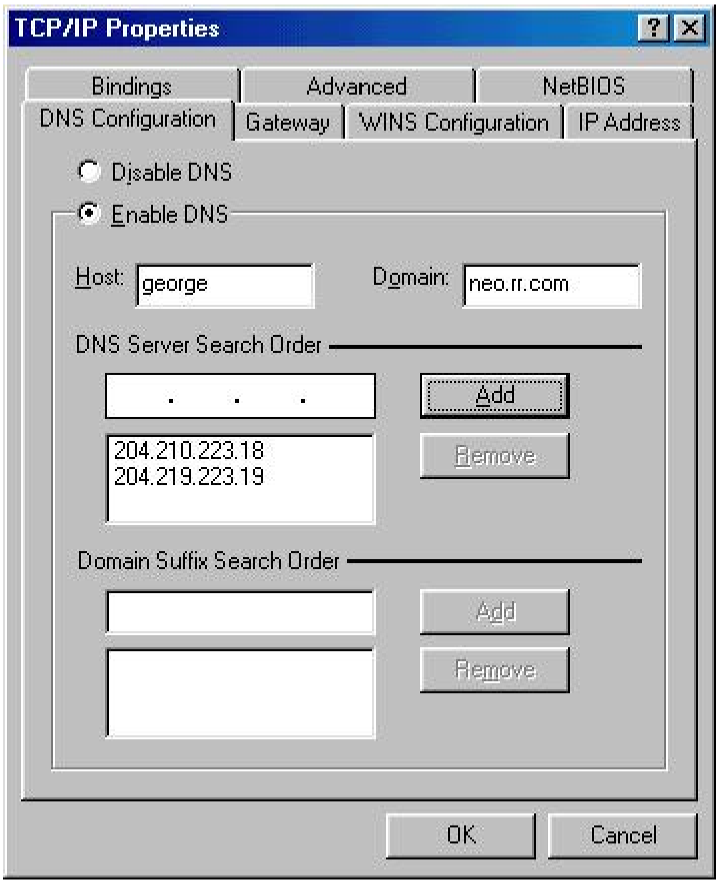
The first gateway in the Installed Gateway list will be the default. The address order in the list will be the order in which these machines are used.

New gateway: . . .

Installed gateways: 192.168.0.1

You should enter the address of your Linux box's internal network card. The address that you will need to enter to use your Coyote gateway is show in the example to the right (192.168.0.1). After entering the address in the "new gateway" field, click the "Add" button.

The only other step to getting your Windows computers to function is to set up the DNS entries. Start by clicking on the "DNS Configuration" tab at the top of the dialog box. The dialog should look like this:



○ Disable DNS

● Enable DNS

Host: george

Domain: neo.rr.com

DNS Server Search Order

204.210.223.18
204.219.223.19

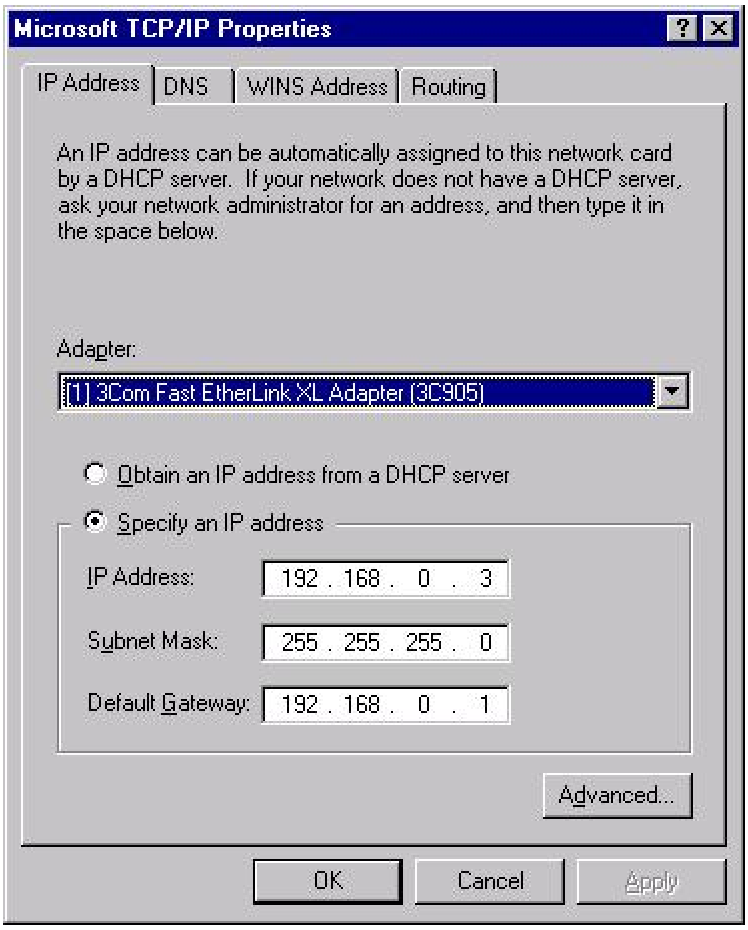
Domain Suffix Search Order

The hostname that appears should be the same as the network name that you chose for your computer. You should change the "Domain" to reflect the domain name that is used by your Internet Service Provider. As for the DNS server IP addresses, you can obtain by using the "Show running configuration" option on the Coyote configuration main menu. Each DNS server for your area should be listed in the "DNS Server Search Order" entries.

### *Windows NT*

The first step is to make sure that you have TCP/IP installed. By default, Windows NT will install TCP/IP, so the installation procedure is not documented here.

To get to the TCP/IP configuration screens, double click the network icon from the control panel. Next, select the protocol tab and select tcp/ip from the list of available protocols. After double clicking on the tcp/ip entry (or clicking the properties button), you should get the "Microsoft TCP/IP Properties" dialog box. This dialog box should appear as follows:



An IP address can be automatically assigned to this network card by a DHCP server. If your network does not have a DHCP server, ask your network administrator for an address, and then type it in the space below.

Adapter: [1] 3Com Fast EtherLink XL Adapter (3C905)

○ Obtain an IP address from a DHCP server

● Specify an IP address

IP Address: 192 . 168 . 0 . 3

Subnet Mask: 255 . 255 . 255 . 0

Default Gateway: 192 . 168 . 0 . 1

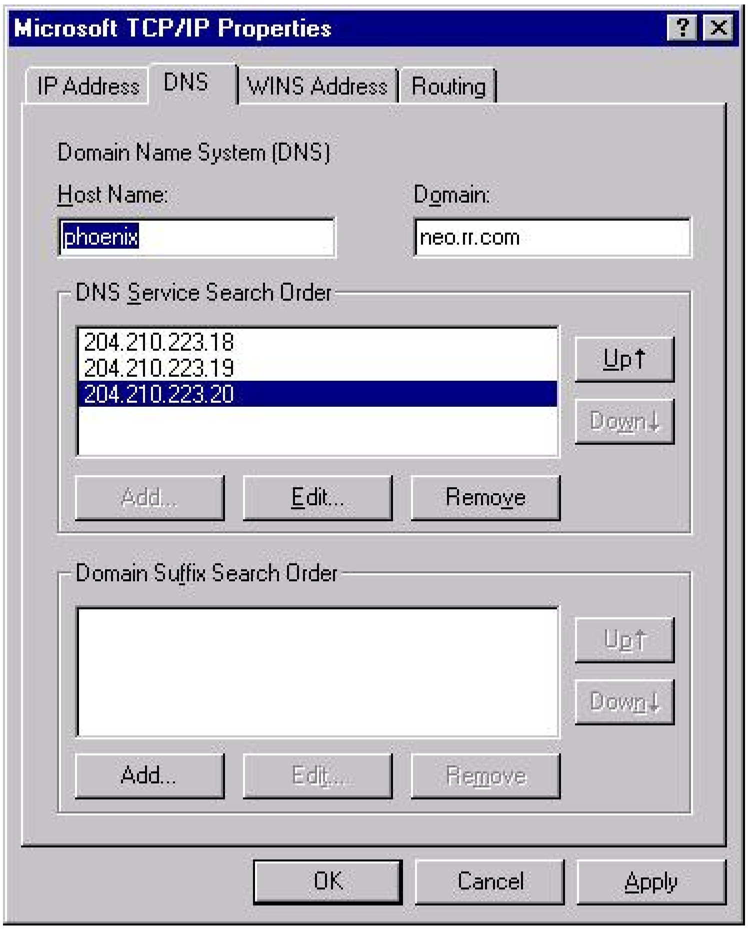
You will note that the "Obtain IP address from DHCP server" is not selected (the default setup if your Windows computer was directly connected to the cable modem). If you chose to use the DHCP server in Coyote, then you will need to make sure that "Obtain IP address from DHCP server" is selected and then your setup is complete. If you selected not to use the DHCP server in Coyote, you will need to specify the address manually.

The settings that are used above should closely reflect those that you will need to use with Coyote.

The only thing that you will need to change from machine to machine will be the "IP Address" entry. These addresses will start at 192.168.0.2 and can be assigned as high as 192.168.0.254. Each machine will need a different IP address.

The subnet mask and gateway should be the same on all machines and should be entered just as above. The Gateway address should read "192.168.0.1" and the netmask should be "255.255.255.0".

The only other step to getting your Windows computers to function is to set up the DNS entries. Start by clicking on the "DNS" tab at the top of the dialog box. The dialog should look like the image below.



Domain Name System (DNS)

Host Name: phoenix

Domain: neo.rr.com

DNS Service Search Order

204.210.223.18
204.210.223.19
204.210.223.20

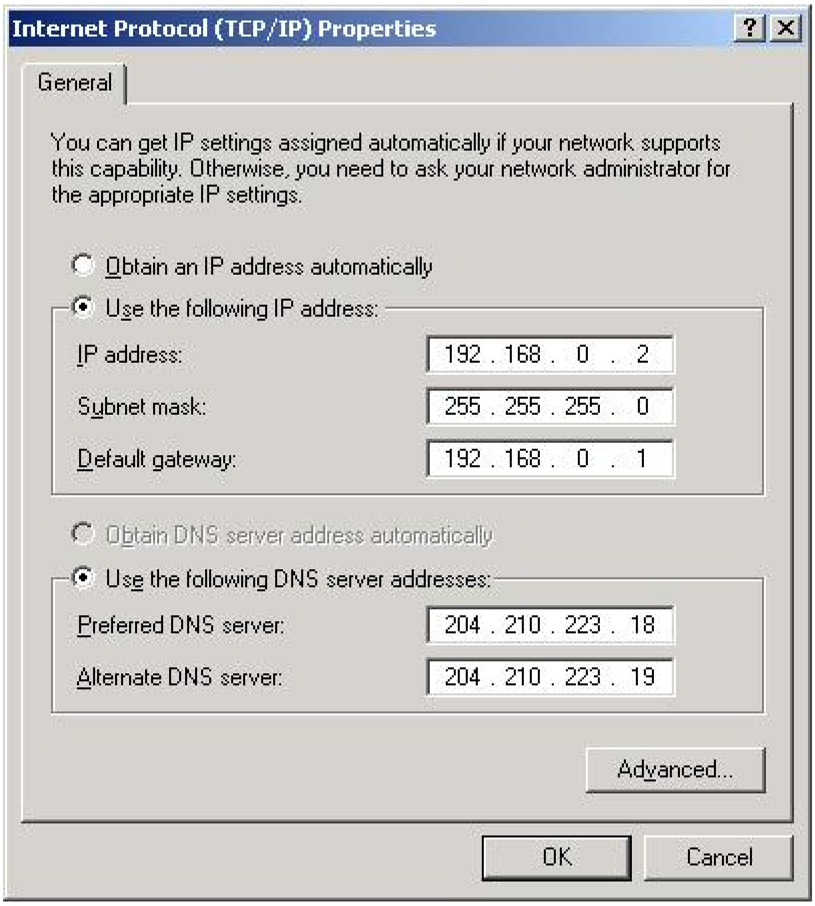
Domain Suffix Search Order

The hostname that appears should be the same as the network name that you chose for your computer. You should change the Domain to reflect the domain name used by your Internet Service Provider. As for the DNS server IP addresses, you can obtain these addresses by using the "Show running configuration" option on the Coyote configuration main menu.

### *Windows 2000*

The first step is to make sure that you have TCP/IP installed. To check to make sure that it is installed, double click on the "Network and Dialup Connections" icon in your control panel. You should see an icon for your "Local Area Network Connection". Right click on this icon and select the properties option from the menu. If TCP/IP is not listed, click the "Install..." button and select "protocol". From the list of available protocols, select TCP/IP and click the OK button. Windows will now install the TCP/IP protocol.

From the Windows networking dialog (mentioned above), double click on the TCP/IP entry. The TCP/IP properties box will look like the following:


Internet Protocol (TCP/IP) Properties

General

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: 192 . 168 . 0 . 2

Subnet mask: 255 . 255 . 255 . 0

Default gateway: 192 . 168 . 0 . 1

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: 204 . 210 . 223 . 18

Alternate DNS server: 204 . 210 . 223 . 19

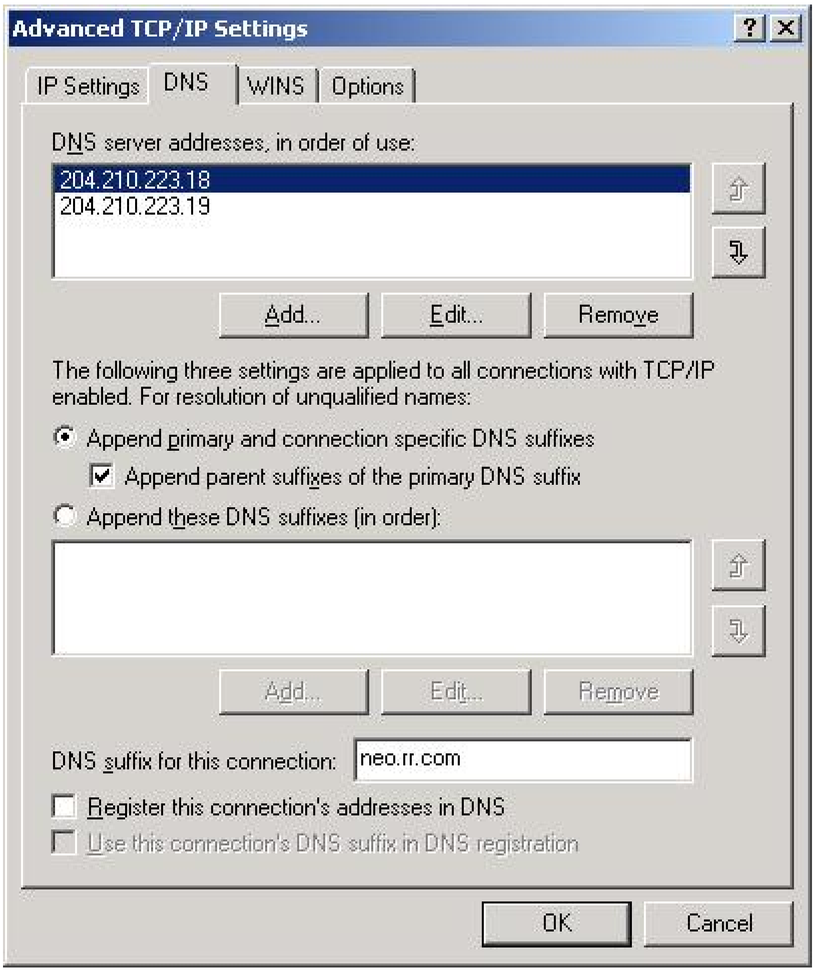
You will note that the "Obtain IP address automatically" is not selected (the default setup if your Windows computer was directly connected to the cable modem). If you have selected to use the DHCP server in Coyote, you will want to check the box "Obtain IP address automatically" and your setup will be complete.

If you chose to not use the DHCP server in Coyote then you will need to specify the address manually. The settings that are used here should closely reflect those that you will need to use.

The only things that you will need to change from machine to machine will be the "IP Address" entry and the DNS servers. The IP addresses will start at 192.168.0.2 and can be assigned as high as 192.168.0.254. Each machine will need a different IP address. The DNS servers will need to reflect the ones used in your area. You can obtain the proper settings for these addresses by using the "Show running configuration" option from the Coyote configuration main menu.

The "default gateway" address and "subnet mask" should be entered as they appear above.

Finally, we need to change a few of the default settings in the "Advanced Options". To open this dialog box, click on the "Advanced" button. Once you get this dialog opened, click on the "DNS" tab. You should now have a dialog that looks like the following:



Advanced TCP/IP Settings

IP Settings | DNS | WINS | Options

DNS server addresses, in order of use:

204.210.223.18
204.210.223.19

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: neo.rr.com

□ Register this connection's addresses in DNS

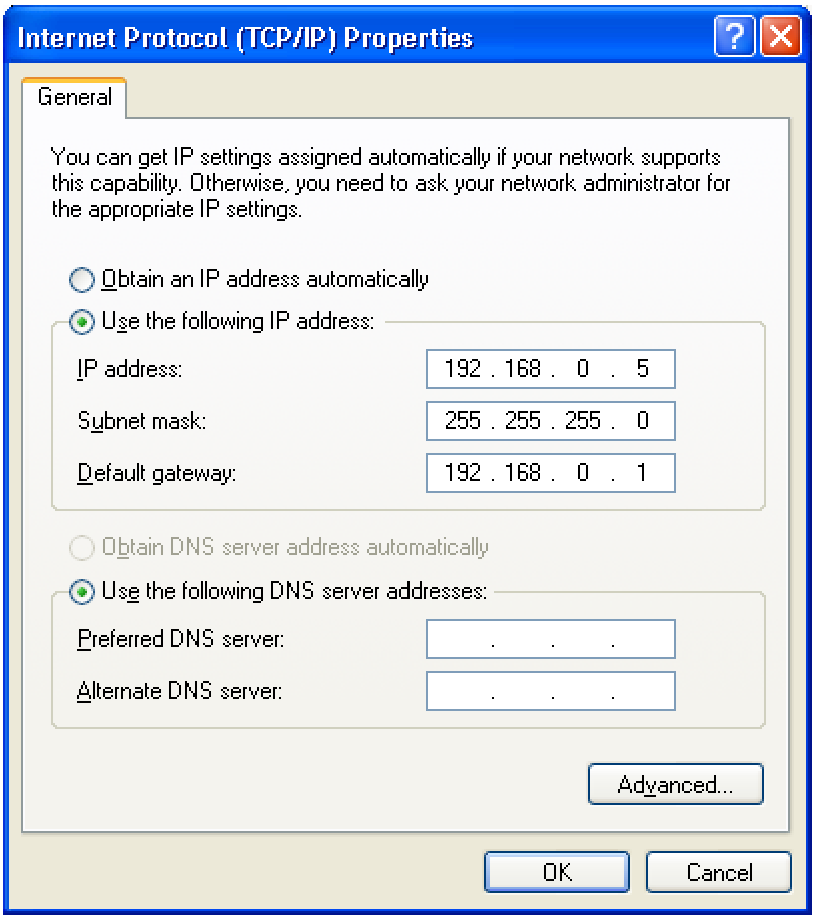
□ Use this connection's DNS suffix in DNS registration

The DNS servers that you specified earlier should appear in the list of server addresses. You should enter the domain name used by your ISP in the "DNS suffix for this connection" edit field. Next, remove the check from the "Register the connection's address in DNS" if it is checked (This option only works with Windows 2000 DNS + Windows 2000 DHCP servers).

### *Windows XP*

The first step is to make sure that you have TCP/IP installed. To check to make sure that it is installed, double click on the "Network and Dialup Connections" icon in your control panel. Then select “Network Connections”. You should see an icon for your "Local Area Network Connection". Right click on this icon and select the properties option from the menu. If TCP/IP is not listed, click the "Install..." button and select "protocol". From the list of available protocols, select TCP/IP and click the OK button. Windows will now install the TCP/IP protocol.

From the Windows networking dialog (mentioned above), double click on the TCP/IP entry. The TCP/IP properties box will look like the following:



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: | 192 . 168 . 0 . 5 |
| Subnet mask: | 255 . 255 . 255 . 0 |
| Default gateway: | 192 . 168 . 0 . 1 |

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

| | |
|---|---|
| Preferred DNS server: | . . . |
| Alternate DNS server: | . . . |

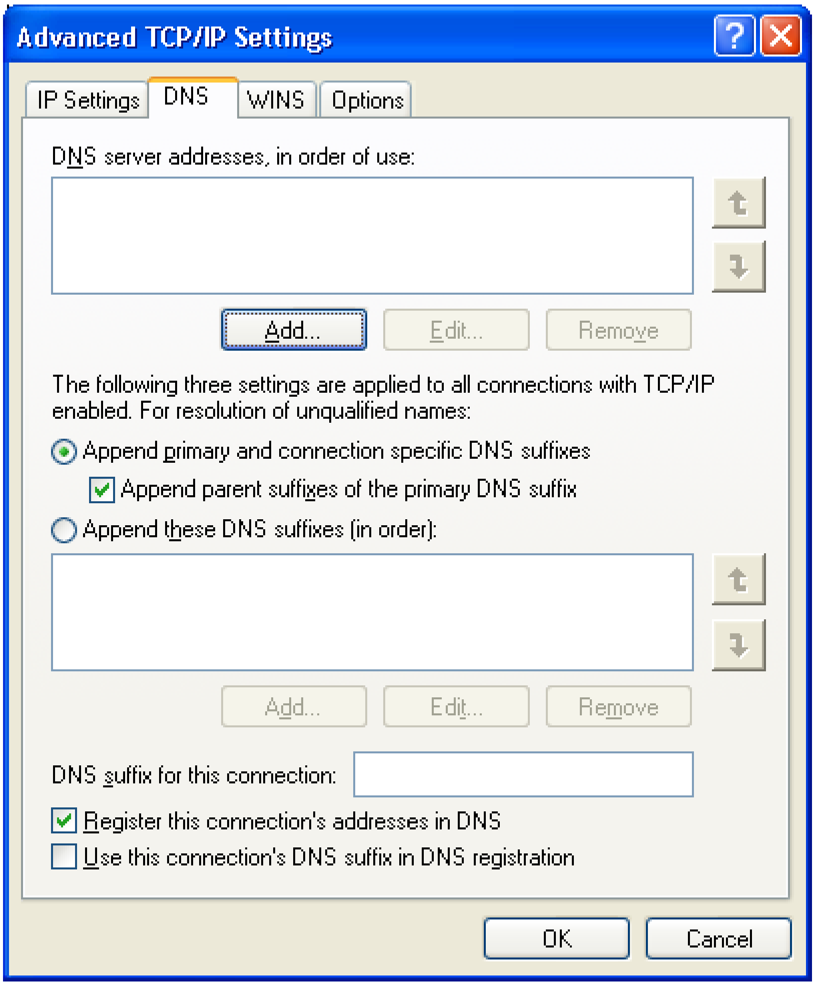
You will note that the "Obtain IP address automatically" is not selected (the default setup if your Windows computer was directly connected to the cable modem). If you have selected to use the DHCP server in Coyote, you will want to check the box "Obtain IP address automatically" and your setup will be complete.

If you chose to not use the DHCP server in Coyote then you will need to specify the address manually. The settings that are used here should closely reflect those that you will need to use.

The only things that you will need to change from machine to machine will be the "IP Address" entry and the DNS servers. The IP addresses will start at 192.168.0.2 and can be assigned as high as 192.168.0.254. Each machine will need a different IP address. The DNS servers will need to reflect the ones used in your area. You can obtain the proper settings for these addresses by using the "Show running configuration" option from the Coyote configuration main menu.

The "default gateway" address and "subnet mask" should be entered as they appear above.

Finally, we need to change a few of the default settings in the "Advanced Options". To open this dialog box, click on the "Advanced" button. Once you get this dialog opened, click on the "DNS" tab. You should now have a dialog that looks like the following:



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

The DNS servers that you specified earlier should appear in the list of server addresses. You should enter the domain name used by your ISP in the "DNS suffix for this connection" edit field. Next, remove the check from the "Register the connection's address in DNS" if it is checked (This option only works with Windows 2000 DNS + Windows 2000 DHCP servers).

## Firewalling with ipchains (from the ipchains man page)

Ipchains is the program that is used to set up, maintain, and inspect the IP firewall rules in the Linux kernel. These rules can be divided into 4 different categories: the IP input chain, the IP output chain, the IP forwarding chain, and user defined chains.

For each of these categories, a separate table of rules is maintained, any of which might refer to one of the user-defined chains.

### *Ipchains Targets*

A firewall rule specifies criteria for a packet, and a target. If the packet does not match, the next rule in the chain is then examined; if it does match, then the next rule is specified by the value of the target, which can be the name of a user-defined chain, or one of the special values ACCEPT, DENY, REJECT, MASQ, REDIRECT, or RETURN.

ACCEPT means to let the packet through. DENY means to drop the packet on the floor. REJECT means the same as drop, but is more polite and easier to debug, since an ICMP message is sent back to the sender indicating that the packet was dropped. (Note that DENY and REJECT are the same for ICMP packets.)

MASQ is only legal for the forward and user defined chains. With this, packets will be masqueraded as if they originated from the local host. Furthermore, reverse packets will be recognized as such and they will be demasqueraded automatically, bypassing the forwarding chain.

With REDIRECT, packets will be redirected to a local socket, even if they were sent to a remote host. If the specified redirection port is 0, which is the default value, the destination port of a packet will be used as the redirection port. When this target is used, an optional extra argument (the port number) can be supplied.

If the end of a user-defined chain is reached, or a rule with target RETURN is matched, then the next rule in the previous (calling) chain is examined. If the end of a built-in chain is reached, or a rule in a built-in chain with target RETURN is matched, the target specified by the chain policy determines the fate of the packet.

### *Ipchains Commands*

These options specify the specific action to perform; only one of them can be specified on the command line, unless otherwise specified below. For all the long versions of the command and option names, you only need to use enough letters to ensure that ipchains can differentiate it from all other options.

-A, --append

Append one or more rules to the end of the selected chain. When the source and/or destination names resolve to more than one address, a rule will be added for each possible address combination.

-D, --delete

Delete one or more rules from the selected chain. There are two versions of this command: the rule can be specified as a number in the chain (starting at 1 for the first rule) or a rule to match.

-R, --replace

Replace a rule in the selected chain. If the source and/or destination names resolve to multiple addresses, the command will fail. Rules are numbered starting at 1.

-I, --insert

Insert one or more rules in the selected chain as the given rule number. So, if the rule number is 1, the rule or rules are inserted at the head of the chain.

-L, --list

List all rules in the selected chain. If no chain is selected, all chains are listed. It is legal to specify the -Z (zero) option as well, in which case no chain may be specified. The exact output is affected by the other arguments given.

-F, --flush

Flush the selected chain. This is equivalent to deleting all the rules one by one.

-Z, --zero

Zero the packet and byte counters in all chains. It is legal to specify the -L, --list (list) option as well, to see the counters immediately before they are cleared; if this is done, then no specific chain can be specified (they will *all* be displayed and cleared).

-N, --new-chain

Create a new user-defined chain of the given name. There must be no target of that name already.

-X, --delete-chain

Delete the specified user-defined chain. There must be no references to the chain (if there are you must delete or replace the referring rules before the chain can be deleted). If no argument is given, it will attempt to delete every non-built-in chain.

-P, --policy

Set the policy for the chain to the given target. See the section TARGETS for the legal targets. Only non-userdefined chains can have policies, and neither built-in nor user-defined chains can be policy targets.

-M, --masquerading

This option allows viewing of the currently masqueraded connections (in conjunction with the -L option) or to set the kernel masquerading parameters (with the -S option).

-S, --set tcp tcpfin udp

Change the timeout values used for masquerading. This command always takes 3 parameters, representing the timeout values (in seconds) for TCP sessions, TCP sessions after receiving a FIN packet, and UDP packets, respectively. A timeout value 0 means that the current timeout value of the corresponding entry is preserved. This option is only allowed in combination with the -M flag.

-C, --check

Check the given packet against the selected chain. This is extremely useful for testing, as the same kernel routines used to check "real" network packets are used to check this packet. It can be used to check user-defined chains as well as the built-in ones. The same arguments used to specify firewall rules are used to construct the packet to be tested. In particular, the -s (source), -d (destination), -p (protocol), and -i (interface) flags are compulsory.

-h, --help

Give a (currently very brief) description of the command syntax. If followed by the word *icmp*, then a list of ICMP names is listed.

-V, --version

Simply output the ipchains version number.

### *Ipchains Parameters*

The following parameters make up a rule specification (as used in the add, delete, replace, append and check commands).

-p, --protocol*[!] protocol*

The protocol of the rule or of the packet to check. The specified protocol can be one of *tcp*, *udp*, *icmp*, or *all*, or it can be a numeric value, representing one of these protocols or a different one. Also a protocol name from /etc/protocols is allowed. A "!" argument before the protocol inverts the test. The number zero is equivalent to *all*. Protocol *all* will match with all protocols and is taken as default when this option is omitted. *All* may not be used in combination with the check command.

-s, --source, --src [!] *address*[/*mask*] [!] [*port[:port]*]

Source specification. *Address* can be either a hostname, a network name, or a plain IP address. The *mask* can be either a network mask or a plain number, specifying the number of 1's at the left side of the network mask. Thus, a mask of *24* is equivalent to *255.255.255.0*. A "!" argument before the address specification inverts the sense of the address.

The source may include a port specification or ICMP type. This can either be a service name, a port number, a numeric ICMP type, or one of the ICMP type names shown by the command

```
ipchains -h icmp
```

Note that many of these ICMP names refer to both a type and code, meaning that an ICMP code after the -d flag is illegal. In the rest of this paragraph, a *port* means either a port specification or an ICMP type. An inclusive range can also be specified, using the format *port*:*port*. If the first port is omitted, "0" is assumed; if the last is omitted, "65535" is assumed.

Ports may only be specified in combination with the *tcp*, *udp*, or *icmp* protocols. A "!" before the port specification inverts the sense. When the check command is specified, exactly one port is required, and if the -f (fragment) flag is specified, no ports are allowed.

--source-port [!] [*port[:port]*]

This allows separate specification of the source port or port range. See the description of the -s flag above for details. The flag --sport is an alias for this option.

-d, --destination, --dst [!] *address*[/*mask*] [!] [*port[:port]*]

Destination specification. See the description of the -s (source) flag for a detailed description of the syntax. For ICMP, which does not have ports, a "destination port" refers to the numeric ICMP code.

--destination-port [!] [*port[:port]*]

This allows separate specification of the ports. See the description of the -s flag for details. The flag --dport is an alias for this option.

--icmp-type [!] typename

This allows specification of the ICMP type (use the -h icmp option to see valid ICMP type names). This is often more convenient than appending it to the destination specification.

-j, --jump *target*

This specifies the target of the rule; i.e. what to do if the packet matches it. The target can be a user-defined chain (not the one this rule is in) or one of the special targets which decide the fate of the packet immediately. If this option is omitted in a rule, then matching the rule will have no effect on the packet's fate, but the counters on the rule will be incremented.

-i, --interface *[!] name*

Optional name of an interface via which a packet is received (for packets entering the input chain), or via which is packet is going to be sent (for packets entering the forward or output chains). When this option is omitted, the empty string is assumed, which has a special meaning and will match with any interface name. When the "!" argument is used before the interface name, the sense is inverted. If the interface name ends in a "+", then any interface which begins with this name will match.

[!] -f, --fragment

This means that the rule only refers to second and further fragments of fragmented packets. Since there is no way to tell the source or destination ports of such a packet (or ICMP type), such a packet will not match any rules which specify them. When the "!" argument precedes the "-f" flag, the sense is inverted.

### *Other ipchains Options*

The following additional options can be specified:

-b, --bidirectional

Bidirectional mode. The rule will match with IP packets in both directions; this has the same effect as repeating the rule with the source & destination reversed. Note that this does NOT mean that if you allow TCP syn packets out, the -b rule will allow non-SYN packets back in: the reverse rule is exactly the same as the rule you entered. This means that it's usually better to simply avoid the -b flag and spell the rules out explicitly.

-v, --verbose

Verbose output. This option makes the list command show the interface address, the rule options (if any), and the TOS masks. The packet and byte counters are also listed, with the suffix 'K', 'M' or 'G' for 1000, 1,000,000 and 1,000,000,000 multipliers respectively (but see the -x flag to change this). When used in combination with -M, information related to delta sequence numbers will also be listed. For appending, insertion, deletion and replacement, this causes detailed information on the rule or rules to be printed.

-n, --numeric

Numeric output. IP addresses and port numbers will be printed in numeric format. By default, the program will try to display them as host names, network names, or services (whenever applicable).

-l, --log

Turn on kernel logging of matching packets. When this option is set for a rule, the Linux kernel will print some information of all matching packets (like most IP header fields) via *printk*().

-o, --output *[maxsize]*

Copy matching packets to the userspace device. This is currently mainly for developers who want to play with firewalling effects in userspace. The optional maxsize argument can be used to limit the maximum number of bytes from the packet which are to be copied. This option is only valid if the kernel has been compiled with CONFIG_IP_FIREWALL_NETLINK set.

-m, --mark *markvalue*

Mark matching packets. Packets can be marked with a 32-bit unsigned value which may (one day) change how they are handled internally. If you are not a kernel hacker you are unlikely to care about this. If the string *markvalue* begins with a + or -, then this value will be added or subtracted from the current marked value of the packet (which starts at zero).

-t, --TOS *andmask xormask*

Masks used for modifying the TOS field in the IP header. When a packet matches a rule, its TOS field is first bitwise and'ed with first mask and the result of this will be bitwise xor'ed with the second mask. The masks should be specified as hexadecimal 8-bit values. As the LSB of the TOS field must be unaltered (RFC 1349), TOS values which would cause it to be altered are rejected, as are any rules which always set more than one TOS bit. Rules which might set multiple TOS bits for certain packets result in warnings (sent to stdout) which can be ignored if you know that packets with those TOS values will never reach that rule. Obviously, manipulating the TOS is a meaningless gesture if the rule's target is *DENY* or *REJECT*.

-x, --exact

Expand numbers. Display the exact value of the packet and byte counters, instead of only the rounded number in K's (multiples of 1000) M's (multiples of 1000K) or G's (multiples of 1000M). This option is only relevant for the -L command.

[!] -y, --syn

Only match TCP packets with the SYN bit set and the ACK and FIN bits cleared. Such packets are used to request TCP connection initiation; for example, blocking such packets coming in an interface will prevent incoming TCP connections, but outgoing TCP connections will be unaffected. This option is only meaningful when the protocol type is set to TCP. If the "!" flag precedes the "-y", the sense of the option is inverted.

--line-numbers

When listing rules, add line numbers to the beginning of each rule, corresponding to that rule's position in the chain.

--no-warnings

Disable all warnings.

A complete explanation of ipchains and firewalling is beyond the scope of this document, for more detailed information see the ipchains How-to. (http://www.netfilter.org/ipchains/HOWTO.html)

To change your Coyote Linux ipchains rules you can enter ipchains commands at the command prompt, you can access the firewall script from the configuration menu, or you can just edit the file /etc/rc.d/rc.firewall.

### *Some Things NOT to Filter*

### ICMP packets

ICMP packets are used (among other things) to indicate failure for other protocols (such as TCP and UDP). `destination-unreachable' packets in particular. Blocking these packets means that you will never get `Host unreachable' or `No route to host' errors; any connections will just wait for a reply that never comes. This is irritating, but rarely fatal.

A worse problem is the role of ICMP packets in MTU discovery. All good TCP implementations (Linux included) use MTU discovery to try to figure out what the largest packet that can get to a destination without being fragmented (fragmentation slows performance, especially when occasional fragments are lost). MTU discovery works by sending packets with the "Don't Fragment" bit set, and then sending smaller packets if it gets an ICMP packet indicating "Fragmentation needed but DF set" (`fragmentation-needed'). This is a type of `destination-unreachable' packet, and if it is never received, the local host will not reduce MTU, and performance will be abysmal or non-existent.

Note that it is common to block all ICMP redirect messages (type 5); these can be used to manipulate routing (although good IP stacks have safeguards), and so are often seen as slightly risky.

### TCP Connections to DNS (nameservers)

If you're trying to block outgoing TCP connections, remember that DNS doesn't always use UDP; if the reply from the server exceeds 512 bytes, the client uses a TCP connection (still going to port number 53) to get the data.

This can be a trap because DNS will `mostly work' if you disallow such TCP transfers; you may experience strange long delays and other occasional DNS problems if you do.

If your DNS queries are always directed at the same external source (either directly by using the nameserver line in /etc/resolv.conf or by using a caching nameserver in forward mode), then you need only allow TCP connections to port domain on that nameserver from the local domain port (if using a caching nameserver) or from a high port (> 1023) if using /etc/resolv.conf.

### *Apply Firewall Changes without rebooting*

With Coyote Linux it is possible to make almost any change active without rebooting your machine. To do this for your firewall rules is fairly easy. As long as you start you ipchains rule set by flushing the chains and setting the defaults you need only to run /etc/rc.d/rc.firewall from the command line. You can do the same thing for your masquerading rules. Coyote, by default, always flushes the auto forward rules when you start masquerading and you could add a flush statement for your port forward rules if you use port forward. The to restart masquerading run /etc/rc.d/rc.masquerade

### *Type Of Service Tweaking*

It is possible to effect how your outgoing packets are processed on their journey by modifying the TOS bit of the IP packets. As each packet passes through the rule set you can have ipchains modify this bit. This sound complicated, but it is rather simple. Use a command like the one below substituting ???? with a value from the table below.

```
ipchains -A output -p tcp -d 0.0.0.0/0.0.0.0 0:1024 -t 0x01 ????
```

| **Type of Service** | **Value for ????** |
|---|---|
| Minimum delay | 0x10 |
| Maximum throughput | 0x08 |
| Maximum reliability | 0x4 |
| Minimum cost | 0x02 |

### *An Example Firewall Script*

```
# ipchains entries go here

# General Setup
#################
ipchains -F input
ipchains -P input DENY
ipchains -F output
ipchains -P output DENY

# Variable Setup
##################
HIPT=1024:65535
INTIF=eth0
INTIP=$LOCAL_IPADDR
INTNT=$LOCAL_NETWORK/$LOCAL_NETMASK
EXTIF=ppp0
EXTIP=`getifaddr $EXTIF`
EXTNT=0.0.0.0/0.0.0.0

# Internal Network
####################
ipchains -A input -i $INTIF -s $INTNT -d $INTIP -j ACCEPT
ipchains -A input -i $INTIF -s $INTNT -d $EXTNT -j ACCEPT
ipchains -A output -i $INTIF -s $INTIP -d $INTNT -j ACCEPT
ipchains -A output -i $INTIF -s $EXTNT -d $INTNT -j ACCEPT

# External Network
####################
ipchains -A input -i $EXTIF -s $INTNT -d $EXTNT -j DENY -l
ipchains -A input -i $EXTIF -p tcp -s $EXTNT 21 -d $EXTIP $HIPT -j ACCEPT
ipchains -A input -i $EXTIF -p tcp -s $EXTNT 25 -d $EXTIP $HIPT -j ACCEPT
ipchains -A input -i $EXTIF -p udp -s $EXTNT 53 -d $EXTIP $HIPT -j ACCEPT
ipchains -A input -i $EXTIF -p tcp -s $EXTNT 80 -d $EXTIP $HIPT -j ACCEPT
ipchains -A input -i $EXTIF -p tcp -s $EXTNT 110 -d $EXTIP $HIPT -j ACCEPT
ipchains -A input -i $EXTIF -p tcp -s $EXTNT 443 -d $EXTIP $HIPT -j ACCEPT
ipchains -A input -j DENY –l
ipchains -A output -i $EXTIF -s $EXTNT -d $INTNT -j DENY –l
ipchains -A output -i $EXTIF -s $INTNT -d $EXTNT -j DENY -l
ipchains -A output -i $EXTIF -s $EXTIP -d $EXTNT -j ACCEPT
ipchains -A output -j DENY -l
```

## Masquerading

IP Masquerading allows a set of machines to invisibly access the Internet via the MASQ gateway, your Coyote Linux gateway. To other machines on the Internet, the outgoing traffic will appear to be from the Coyote Linux server itself. In addition to the added functionality, IP Masquerade provides the foundation to create a heavily secured networking environment. This is the portion of Linux that provides NAT-like functionality (NAT= Network Address Translation). This is what gives you ability to have many computers on your LAN, using a single IP address, access the internet simultaneously.

You would also use masquerading if you wanted to have machines from the internet access a machine, or many machines, inside your LAN. Using masquerading the internal machines do not have to have dedicated public IP addresses.

Masqueraded connections will time out after a certain period of time, this is normal behavior. The default settings are good for most people. If you are sure you know what you are doing you can use the following command to specify you own settings:

```
coyote# ipchains -M -S 7200 10 60
```

This command would give you a 2 hour (7200 second) timeout for TCP sessions, a 10 second timeout for traffic after the TCP/IP "FIN" packet is received, and a 60 second timeout for UDP traffic (MASQ'ed ICQ users must enable a 30sec, firewall timeout in ICQ itself). You do not need to run this command, as these are the default settings.

For a more detailed explanation of IP Masquerading for Linux see the How-to at the Linux Documentation Project. (http://www.linuxdoc.org/HOWTO/IP-Masquerade-HOWTO.html)

### *Coyote Linux Masquerading Modules*

| Module | Purpose |
|---|---|
| ip_masq_autofw | Needed for protocols which are not yet supported by own ip_masq modules |
| ip_masq_cuseeme | Needed for the program CU-SeeMe |
| ip_masq_dplay | Needed for using Microsoft DirectPlay |
| ip_masq_ftp | Needed for ftp PASV (passive mode) transfers |
| ip_masq_h323 | Needed for Netmeeting, Intel Internet Phone, other H.323 applications |
| ip_masq_icq | Needed for running ICQ |
| ip_masq_ipsec | Needed for IP Security Protocol support (VPNs) |
| ip_masq_irc | Needed for running IRC |
| ip_masq_mms | Needed for Microsoft Messenger Service |
| ip_masq_mfw | Needed for use of "firewall marks" (for more info: man ipchains(8), option -m) |
| ip_masq_portfw | Needed for redirecting incoming requests to your internal servers |
| ip_masq_pptp | Needed for Point to Point Tunneling Protocol support (VPNs) |
| Ip_masq_quake | Needed for playing Quake 3 Arena across the internet |
| ip_masq_raudio | Needed for Real Audio Player communications |

| | |
|---|---|
| ip_masq_vdolive | Needed for the program VDOLive |

### *Modifying Masquerade Modules*

To list the loaded masquerading modules use the lsmod command. If you see some modules that you are not using you can unload them using the rmmod command. To reload them there is the insmod command.

To permanently remove or add a module you need to either remove it from or add it to the /etc/modules file.

### *Forwarding Requests from the internet to Internal Servers*

If you wanted to run a web server from your LAN you could use a command like this:

```
ipmasqadm autofw -A -r tcp 80 80 -h 192.168.0.2
```

Coyote has a nice feature for adding entries like this to your system permanently. You can add them into the file /etc/coyote/portworwards. You may have to create this file if it doesn't already exist. The format of the file should be something similar to this:

```
# Coyote port auto-forwards

startingport1 endingport1 protocol2 destination2
startingport2 endingport2 protocol2 destination2
```

Remember that using this method will not automatically forward internal requests for you server to the server. If you own a domain name and point it at your external IP which is in turn forwarded to a server inside your firewall, you likely won't be able to access you're your server by name using the normal methods.

This method will only forward the port from the Coyote Linux machine to the internal server, You can not forward one port number to another port number using this method.

### Using differing ports

If your ISP blocks some common server ports such as 21 (ftp) or 80 (http) you may want people to contact you using a different port yet still have your server respond on its intended port. To do this you will need to use port forwarding. Here is the command to accomplish this:

```
ipmasqadm portfw -a -P tcp -L extern_ip extern_port -R internal_ip internal_port
```

You should use `getifaddr eth1` or `getifaddr ppp0` to get your external ip address.

You can add these entries to the end of your /etc/rc.d/rc.masquerade file.

## General System Administration Items

### *Assign a static IP via DHCP*

You can follow this procedure if you want dhcpd to always assign a particular machine a given IP address.

First you need to know the MAC or hardware address of the machine to receive the same IP number. Here are some commands to discover your MAC address:

| OS | Command | Look for… |
|---|---|---|
| Linux | ifconfig | HWaddr |
| Windows 98/ME | winipcfg | Adapter Address |
| Windows NT/2K/XP | ipconfig /all | Physical Address |

Add an entry in the following format to the file /var/lib/lrpkg/dhcpd.con (or /etc/dhcpd.conf).

```
host HostName { hardware ethernet Your-HostMAC; fixed-address Desired-IP; }
```

Example:

```
host server01 { hardware 00:00:00:00:00:00; fixed-address 192.168.0.2; }
```

The fixed address you choose should NOT be in your normal DHCP scope. In other words, you can not reassign a number that falls between your DHCPD_START and your DHCPD_END.

### *Backing up Coyote*

Follow this procedure to backup Coyote Linux from the Coyote Linux machine.

```
coyote# dd if=/dev/fd0 of=/root/coyote.img
```

Take the Coyote Floppy out of the drive and replace it with a blank one. Then use the next command to write this to another floppy:

```
coyote# dd if=/root/coyote.img of=/dev/fd0
```

### *Change the MAC address of the NICs on the Coyote router*

Run this command:

```
coyote# ifconfig eth0 hw ether 00:00:00:00:00:00
```

### *Changing the RAM disk size*

First, you have to mount your floppy disk. (`mount /de/fd0 /mnt`)

Next, edit syslinux.cfg. (`ae /mnt/syslinux.cfg`)

Then, look for `ramdisk_size=4096`, set this to the desired size in Megabytes multiplied by 1024. For example, for an 8 Megabyte RAM disk set `ramdisk_size=8192`.

Save the file and exit.

Finally unmount the floppy: (`umount /dev/fd0`)

### *Change webadmin port*

If you would like to change the port your webadmin server runs on from the default 8180, you can edit the file /etc/rc.d/pkgs/rc.webadmin, find 8180 and replace that with the port you want to use.

### *Check which IPs have been given out, and to who*

To accomplish this you need to run the following command at the command prompt and browse the log file.

```
coyote# ae /var/state/dhcp/dhcp.leases
```

### *Determine the IP assigned to you by your ISP*

Simply type this command:

```
coyote# getifaddr ppp0
```

### *Disable a Network Interface*

To temporarily (or permanently) disable one interface do the following:

```
coyote# ifconfig the-interface-name down
```

To bring it back up:

```
coyote# ifconfig the-interface-name up
```

### *Escape Codes (Terminal Multimedia)*

This section contains information that will probably not be needed by most users, except for the screensaver settings. This section could provide useful, however, to anyone doing shell scripting for Coyote Linux.

Using escape codes it is possible things with Coyote Linux that would otherwise not be possible. You can enter escape codes in two different ways. First at the console you can hit the escape key and then enter the code, or you could use `echo` to echo the escape key and the code. The latter is the method I will be using because it is an all around easier method.

Use caution when entering escape codes as your terminal can get seriously screwed up!

Here is a sample escape sequence with explanation of how it breaks down:

```
coyote# echo -e "\033[9;0]"
```

| | |
|---|---|
| echo –e | Echo using escape codes. You need to use quotes when using –e. |
| \033 | This is the escape character, ASCII 033 (\33 will also work) |
| [9;0] | The code to escape. The format of these vary. |

### Change Screensaver Settings

You can change the console screensaver timeout using the escape codes [9;x], where x would represent the screensaver timeout period and zero means disable the screensaver altogether. The following command will set the screensaver timeout to thirty (30) minutes:

```
coyote# echo -e "\033[9;30]"
```

This command will disable the screensaver completely:

```
coyote# echo -e "\033[9;0]"
```

Add these commands to your /etc/rc.d/rc.local file to make them persistent across boots.

### Reset the Terminal

Sometimes terminals can go haywire (especially if you try experimenting with options in this chapter) and start displaying odd charcters, this command will reset a terminal in this state:

```
coyote# echo -e "\033c"
```

### Setting Colors

Changing the screen color is really only useful in scripts or for colorizing your prompt. The following command is really a VT100 or Linux terminal escape code, but it changes the current text colors:

```
coyote# echo -e "\033[x;y;zm"
```

The x represents the text attribute, the y represents the foreground color, and z represents the background color.

| X value | Text type | Y value | Text color | Z value | Back Color |
|---|---|---|---|---|---|
| 0 | Reset | 30 | Black | 40 | Black |
| 1 | Bright | 31 | Red | 41 | Red |
| 2 | Dim | 32 | Green | 42 | Green |
| 4 | Underscore | 33 | Yellow | 43 | Yellow |
| 5 | Blink | 34 | Blue | 44 | Blue |
| 7 | Reverse | 35 | Magenta | 45 | Magenta |
| 8 | Hidden | 36 | Cyan | 46 | Cyan |
| 3,6 | (unused) | 37 | White/Grey | 47 | White/Grey |

### Setting the Cursor

If you desire, you can change how your cursor appears using a command like this:

```
coyote# echo -e "\033[?x;y;zc"
```

X represents the cursor type, Y represents the cursor foreground color, Z represents the cursor background color. The X value should be between 17 and 22. The Y and Z value can vary between 0 and 255. I am not completely sure on these numbers, you will have to adjust the numbers to taste.

### Sounding the Bell

If, for some reason, you wanted to sound the bell you could use this command:

```
coyote# echo -e "\007"
```

To make things more interesting you could change the bell tone by using the following command. X represents the bell tone in hertz and y represents the duration in milliseconds.

```
coyote# echo -e "\033[10;x]\033[11;y]"
```

If you were musically inclined you could create a whole song, or maybe just a login tune like some other operating systems have. Here is how the hertz value translates into musical notes.

| Note | Hertz | Note | Hertz | Note | Hertz | Note | Hertz |
|---|---|---|---|---|---|---|---|
| C | 65.41 | F# | 185.00 | C | 523.25 | F# | 1479.98 |
| C# | 69.30 | G | 196.00 | C# | 554.37 | G | 1567.98 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| D | 73.42 | G# | 207.65 | D | 587.33 | G# | 1661.22 |
| D# | 77.78 | A | 220.00 | D# | 622.25 | A | 1760.00 |
| E | 82.41 | A# | 233.08 | E | 659.26 | A# | 1864.66 |
| F | 87.31 | B | 246.94 | F | 698.46 | B | 1975.53 |
| F# | 92.50 | C | 261.63 | F# | 739.99 | C | 2093.00 |
| G | 98.00 | C# | 277.18 | G | 783.99 | C# | 2217.46 |
| G# | 103.83 | D | 293.66 | G# | 830.61 | D | 2349.32 |
| A | 110.00 | D# | 311.13 | A | 880.00 | D# | 2489.02 |
| A# | 116.54 | E | 329.63 | A# | 932.33 | E | 2637.02 |
| B | 123.47 | F | 349.23 | B | 987.77 | F | 2793.83 |
| C | 130.81 | F# | 369.99 | C | 1046.50 | F# | 2959.96 |
| C# | 138.59 | G | 392.00 | C# | 1108.73 | G | 3135.96 |
| D | 146.83 | G# | 415.30 | D | 1174.66 | G# | 3322.44 |
| D# | 155.56 | A | 440.00 | D# | 1244.51 | A | 3520.00 |
| E | 164.81 | A# | 466.16 | E | 1328.51 | A# | 3729.31 |
| F | 174.61 | B | 493.88 | F | 1396.91 | B | 3951.07 |

Once you are done you can disable the beeper again like this:

```
coyote# echo –e “\033[10;0]\033[11;0]”
```

### Other Console Tricks

Here are some other, mostly useless, console tricks.

| Command | Result |
|---|---|
| echo -e “\33[2l” | Lock keyboard  (be careful with this) |
| echo -e “\33[2h” | Unlock keyboard |
| echo -e “\33[4l” | Insert mode |
| echo -e “\33[4h” | Replace mode (has some bad effects!) |
| echo -e “\33[20l” | CR+LF mode (Microsoft-like, useless mode) |
| echo -e “\33[20h” | LF mode |
| echo -e “\33[?1l” | Cursor keys application mode |
| echo -e “\33[?1h” | Cursor keys normal mode |
| echo -e “\33[?2l” | ANSI mode |
| echo -e “\33[?2h” | VT52 mode |
| echo -e “\33[?3l” | 132 column screen width (doesn’t work at console) |
| echo -e “\33[?3h” | 80 column screen width |
| echo -e “\33[?4l” | Smooth scroll |
| echo -e “\33[?4h” | Jump scroll |
| echo -e “\33[?5l” | Invert screen colors |
| echo -e “\33[?5h” | Un-invert screen colors |
| echo -e “\33[?6l” | Set relative coordinates |
| echo -e “\33[?6h” | Set absolute coordinates |

| | |
|---|---|
| echo -e “\33[?7l” | Auto wrap on |
| echo -e “\33[?7h” | Auto wrap off |
| echo -e “\33[?8l” | Auto repeat on |
| echo -e “\33[?8h” | Auto repeat off |
| echo -e “\33[xA” | Cursor up by x number of lines |
| echo -e “\33[xB” | Cursor down by x number of lines |
| echo -e “\33[xC” | Cursor left x number of columns |
| echo -e “\33[xD” | Cursor right by x number of columns |
| echo -e “\33[y;xH” | Set cursor position to x,y |
| echo -e “\33[y;xf” | Set cursor position to x,y |
| echo -e “\33D” | Index (cursor down with scroll up when at margin) |
| echo -e “\33M” | Reverse index (cursor up with scroll down when at margin) |
| echo -e “\33E” | Next line (CR+Index) |
| echo -e “\337” | Save cursor and attribute |
| echo -e “\338” | Restore cursor and attribute |
| echo -e “\33H” | Set horizontal tab |
| echo -e “\33[g” | Clear horizontal tab |
| echo -e “\33#3” | Double-wide, double-high text, upper portion |
| echo -e “\33#4” | Double-wide, double-high text, lower portion |
| echo -e “\33#5” | Single-width, single-height text (normal text) |
| echo -e “\33#6” | Double-width, single-height text |
| echo -e “\33[c” | Request terminal ID (6c or \33[?6c returned for Linux or VT102) |
| echo -e “\33#8” | Screen adjustment (I don’t know what this is, fills the screen with Es) |

### Remote Syslog-ing

To enable remote logging of syslog messages add this line in /etc/syslog.conf:

```
*.*     @logger-server-ip
```

### Using Other LRP Packages

It is possible to use some general LRP packages with Coyote Linux, some experimentation on your part will be required though as not all LRP packages will work. This procedure will show you how to install the package so that it (hopeful) loads upon boot. You can even do this from your Coyote Linux machine. I will give an example of how to do this using your Coyote machine, for all the Windows users out there.

First, copy your LRP to a floppy and then stick it into the Coyote linux machine and copy it off the floppy:

```
mkdir /tmp/lrp
mount /dev/fd0 /mnt
cp *.lrp /tmp/lrp
```

```
mv /tmp/lrp/*.lrp /tmp/lrp/*.tgz
umount /mnt
```

Then, mount the floppy, copy the root.tgz file to your home directory, and unpack it (You may not have to do all this unpacking and repacking, it may be possible to simply add you entry to /mnt/packages on your boot floppy. I have not tested this yet though):

```
cd
mount /dev/fd0 /mnt
cp /mnt/root.tgz ~/
tar –xzvf root.tgz
```

Now we just need to edit ~/var/lib/lrpkg/root.packages and add the packes we intend to install (one per line, leaving one blank line at the end).

Next, we need to save our changes back to the floppy:

```
cd
rm ~/root.tgz
tar -czvf ~/root.tgz *
rm /mnt/root.tgz
mv ~/root.tgz /mnt/
mv /tmp/lrp/* /mnt/
umount /mnt
```

Now when you reboot your Coyote Linux machine the LRP package should load up. Remember, there is guarantee that it will work properly.

## Coyote Linux Command Reference

### *ae*

ae FILE
Edit the specified file. (for more information see: *The MicroEditor e3*)

### *arp*

Return information regarding to the Address Resolution Protocol.

### *basename*

basename FILE [SUFFIX]
Strips directory path and suffixes from FILE. If specified, also removes any trailing SUFFIX.
Example:

```
        $ basename /usr/local/bin/foo
        foo
        $ basename /usr/local/bin/
        bin
        $ basename /foo/bar.txt .txt
        bar
```

### *cat*

cat [FILE]...
Concatenates `FILE(s)` and prints them to stdout.
Example:

```
        $ cat /proc/uptime
        110716.72 17.67
```

### *chgrp*

chgrp [OPTION]... GROUP FILE...
Change the group membership of each FILE to GROUP.
Options:

```
        -R      Changes files and directories recursively.
```

Example:

```
        $ ls -l /tmp/foo
        -r--r--r--    1 andersen andersen        0 Apr 12 18:25 /tmp/foo
        $ chgrp root /tmp/foo
        $ ls -l /tmp/foo
        -r--r--r--    1 andersen root            0 Apr 12 18:25 /tmp/foo
```

### *chmod*

chmod [**-R**] MODE[,MODE]... FILE...
Each MODE is one or more of the letters ugoa, one of the symbols +-= and one or more of the letters rwxst.
Options:

```
        -R      Changes files and directories recursively.
```

Example:

```
        $ ls -l /tmp/foo
        -rw-rw-r--    1 root     root            0 Apr 12 18:25 /tmp/foo
        $ chmod u+x /tmp/foo
        $ ls -l /tmp/foo
        -rwxrw-r--    1 root     root            0 Apr 12 18:25 /tmp/foo*
        $ chmod 444 /tmp/foo
        $ ls -l /tmp/foo
        -r--r--r--    1 root     root            0 Apr 12 18:25 /tmp/foo
```

### *chown*

chown [ **-Rh** ]... OWNER[<.|:>[GROUP]] FILE...
Change the owner and/or group of each FILE to OWNER and/or GROUP.
Options:

```
        -R      Changes files and directories recursively.
        -h      Do not dereference symbolic links.
```

Example:

```
        $ ls -l /tmp/foo
        -r--r--r--    1 andersen andersen        0 Apr 12 18:25 /tmp/foo
        $ chown root /tmp/foo
        $ ls -l /tmp/foo
        -r--r--r--    1 root     andersen        0 Apr 12 18:25 /tmp/foo
        $ chown root.root /tmp/foo
        ls -l /tmp/foo
        -r--r--r--    1 root     root            0 Apr 12 18:25 /tmp/foo
```

### *chroot*

chroot NEWROOT [COMMAND...]
Run COMMAND with root directory set to NEWROOT.
Example:

```
        $ ls -l /bin/ls
        lrwxrwxrwx    1 root     root           12 Apr 13 00:46 /bin/ls ->
/BusyBox
        $ mount /dev/hdc1 /mnt -t minix
        $ chroot /mnt
        $ ls -l /bin/ls
        -rwxr-xr-x    1 root     root        40816 Feb  5 07:45 /bin/ls*
```

### *clear*

clear
Clear screen.

### *cp*

cp [OPTION]... SOURCE DEST
Copies SOURCE to DEST, or multiple `SOURCE(s)` to DIRECTORY.

```
        -a      Same as -dpR
        -d      Preserves links
        -p      Preserves file attributes if possible
        -f      force (implied; ignored) - always set
        -R      Copies directories recursively
```

### *ctar*

ctar [OPTION] FILE –X FILES

Options:

-x, --exclude exclude files

### *cut*

cut [OPTION]... [FILE]...

Prints selected fields from each input FILE to standard output.

Options:

```
-b LIST         Output only bytes from LIST
-c LIST         Output only characters from LIST
-d CHAR         Use CHAR instead of tab as the field delimiter
-s              Output only the lines containing delimiter
-f N            Print only these fields
-n              Ignored
```

Example:

```
$ echo "Hello world" | cut -f 1 -d ' '
Hello
$ echo "Hello world" | cut -f 2 -d ' '
world
```

### *date*

date [OPTION]... [+FORMAT]

Displays the current time in the given FORMAT, or sets the system date.

Options:

```
-R              Outputs RFC-822 compliant date string
-d STRING       display time described by STRING, not `now'
-s              Sets time described by STRING
-u              Prints or sets Coordinated Universal Time
```

Example:

```
$ date
Wed Apr 12 18:52:41 MDT 2000
```

### *dd*

dd [if=FILE] [of=FILE] [bs=N] [count=N]
[skip=N]
[seek=N]
[conv=notrunc|noerror|sync]

Copy a file, converting and formatting according to options

```
if=FILE         read from FILE instead of stdin
of=FILE         write to FILE instead of stdout
bs=N            read and write N bytes at a time
count=N         copy only N input blocks
skip=N          skip N input blocks
seek=N          skip N output blocks
conv=notrunc    don't truncate output file
conv=noerror    continue after read errors
conv=sync       pad blocks with zeros
```

Numbers may be suffixed by c (x1), w (x2), b (x512), kD (x1000), k (x1024), MD (x1000000), M (x1048576), GD (x1000000000) or G (x1073741824).

Example:

```
$ dd if=/dev/zero of=/dev/ram1 bs=1M count=4
4+0 records in
```

```
        4+0 records out
```

### *df*

df [**-hmk**] [FILESYSTEM ...]
Print the filesystem space used and space available.
Options:

```
        -h      print sizes in human readable format (e.g., 1K 243M 2G )
        -m      print sizes in megabytes
        -k      print sizes in kilobytes(default)
```

Example:

```
        $ df
        Filesystem           1k-blocks      Used Available Use% Mounted on
        /dev/sda3              8690864   8553540    137324  98% /
        /dev/sda1                64216     36364     27852  57% /boot
        $ df /dev/sda3
        Filesystem           1k-blocks      Used Available Use% Mounted on
        /dev/sda3              8690864   8553540    137324  98% /
```

### *dirname*

dirname [FILENAME ...]
Strips non-directory suffix from FILENAME
Example:

```
        $ dirname /tmp/foo
        /tmp
        $ dirname /tmp/foo/
        /tmp
```

### *dmesg*

dmesg [**-c**] [**-n** LEVEL] [**-s** SIZE]
Prints or controls the kernel ring buffer
Options:

```
        -c              Clears the ring buffer's contents after printing
        -n LEVEL        Sets console logging level
        -s SIZE         Use a buffer of size SIZE
```

### *du*

du [**-lsxhmk**] [FILE]...
Summarizes disk space used for each FILE and/or directory. Disk space is printed in units of 1024 bytes.
Options:

```
        -l      count sizes many times if hard linked
        -s      display only a total for each argument
        -h      print sizes in human readable format (e.g., 1K 243M 2G )
        -m      print sizes in megabytes
        -x      skip directories on different filesystems
        -k      print sizes in kilobytes(default)
```

Example:

```
        $ du
        16      ./CVS
        12      ./kernel-patches/CVS
        80      ./kernel-patches
```

```
12      ./tests/CVS
36      ./tests
12      ./scripts/CVS
16      ./scripts
12      ./docs/CVS
104     ./docs
2417    .
```

### *e3*

e3 FILE<br>
Edit the specified file. (for more information see: *The MicroEditor e3*)

### *e3em*

e3 FILE<br>
The e3 editor with EMACS like keybinfings. (for more information see: *The MicroEditor e3*)

### *e3ne*

e3 FILE<br>
The e3 editor with NEdit like keybinfings. (for more information see: *The MicroEditor e3*)

### *e3pi*

e3 FILE<br>
The e3 editor with Pico like keybinfings. (for more information see: *The MicroEditor e3*)

### *e3vi*

e3 FILE<br>
The e3 editor with vi like keybinfings. (for more information see: *The MicroEditor e3*)

### *e3ws*

e3 FILE<br>
The e3 editor with Wordstar like keybinfings. (for more information see: *The MicroEditor e3*)

### *echo*

echo [**-neE**] [ARG ...]

Prints the specified ARGs to stdout

Options:

```
-n      suppress trailing newline
-e      interpret backslash-escaped characters (i.e., \t=tab)
-E      disable interpretation of backslash-escaped characters
```

Example:

```
$ echo "Erik is cool"
Erik is cool
$  echo -e "Erik\nis\ncool"
Erik
is
cool
$ echo "Erik\nis\ncool"
Erik\nis\ncool
```

### *env*

env [**-iu**] [-] [name=value]... [command]
Prints the current environment or runs a program after setting up the specified environment.
Options:

```
        -, -i   start with an empty environment
        -u      remove variable from the environment
```

### *exit*

exit
Logs you out of Coyote Linux.

### *expr*

expr EXPRESSION
Prints the value of EXPRESSION to standard output.
EXPRESSION may be:

```
        ARG1 |  ARG2    ARG1 if it is neither null nor 0, otherwise ARG2
        ARG1 &  ARG2    ARG1 if neither argument is null or 0, otherwise 0
        ARG1 <  ARG2    ARG1 is less than ARG2
        ARG1 <= ARG2    ARG1 is less than or equal to ARG2
        ARG1 =  ARG2    ARG1 is equal to ARG2
        ARG1 != ARG2    ARG1 is unequal to ARG2
        ARG1 >= ARG2    ARG1 is greater than or equal to ARG2
        ARG1 >  ARG2    ARG1 is greater than ARG2
        ARG1 +  ARG2    arithmetic sum of ARG1 and ARG2
        ARG1 -  ARG2    arithmetic difference of ARG1 and ARG2
        ARG1 *  ARG2    arithmetic product of ARG1 and ARG2
        ARG1 /  ARG2    arithmetic quotient of ARG1 divided by ARG2
        ARG1 %  ARG2    arithmetic remainder of ARG1 divided by ARG2
        STRING : REGEXP            anchored pattern match of REGEXP in
STRING
        match STRING REGEXP        same as STRING : REGEXP
        substr STRING POS LENGTH   substring of STRING, POS counted from 1
        index STRING CHARS         index in STRING where any CHARS is found,
                                   or 0
        length STRING              length of STRING
        quote TOKEN                interpret TOKEN as a string, even if
                                   it is a keyword like `match' or an
                                   operator like `/'
        ( EXPRESSION )             value of EXPRESSION
```

Beware that many operators need to be escaped or quoted for shells. Comparisons are arithmetic if both ARGs are numbers, else lexicographical. Pattern matches return the string matched between \( and \) or null; if \( and \) are not used, they return the number of characters matched or 0.

### *false*

false
Return an exit code of FALSE (1).
Example:

```
        $ false
        $ echo $?
        1
```

### ***find***

find [PATH...] [EXPRESSION]
Search for files in a directory hierarchy. The default PATH is the current directory; default EXPRESSION is '**-print**'
EXPRESSION may consist of:

```
        -follow         Dereference symbolic links.
        -name PATTERN   File name (leading directories removed) matches
PATTERN.
        -print          Print (default and assumed).
        -type X         Filetype matches X (where X is one of: f,d,l,b,c,...)
        -perm PERMS     Permissions match any of (+NNN); all of (-NNN);
                        or exactly (NNN)
        -mtime TIME     Modified time is greater than (+N); less than (-N);
                        or exactly (N) days
        -newer FILE     Modified time is more recent than FILE's
```

Example:

```
        $ find / -name /etc/passwd
        /etc/passwd
```

### ***free***

free
Displays the amount of free and used system memory
Example:

```
        $ free
                 total    used     free     shared   buffers
          Mem:   257628   248724   8904     59644    93124
         Swap:   128516   8404     120112
        Total:   386144   257128   129016
```

### ***getifaddr***

getifaddr INTERFACE [OPTION}
Options:

-b
-m

### ***grep***

grep [**-ihHnqvs**] PATTERN [FILEs...]
Search for PATTERN in each FILE or standard input.
Options:

```
        -H      prefix output lines with filename where match was found
        -h      suppress the prefixing filename on output
        -i      ignore case distinctions
        -l      list names of files that match
        -n      print line number with output lines
        -q      be quiet. Returns 0 if result was found, 1 otherwise
        -v      select non-matching lines
        -s      suppress file open/read error messages
```

Example:

```
        $ grep root /etc/passwd
        root:x:0:0:root:/root:/bin/bash
        $ grep ^[rR]oo. /etc/passwd
        root:x:0:0:root:/root:/bin/bash
```

### gunzip

gunzip [OPTION]... FILE
Uncompress FILE (or standard input if FILE is '-').
Options:

```
        -c      Write output to standard output
        -t      Test compressed file integrity
```

Example:

```
        $ ls -la /tmp/BusyBox*
        -rw-rw-r-- 1 andersen andersen 557009 Apr 11 10:55 /tmp/01.tar.gz
        $ gunzip /tmp/BusyBox-0.43.tar.gz
        $ ls -la /tmp/BusyBox*
        -rw-rw-r-- 1 andersen andersen  1761280 Apr 14 17:47 /tmp/01.tar
```

### gzip

gzip [OPTION]... FILE
Compress FILE with maximum compression. When FILE is '-', reads standard input. Implies **-c**.
Options:

```
        -c      Write output to standard output instead of FILE.gz
        -d      decompress
```

Example:

```
        $ ls -la /tmp/busybox*
        -rw-rw-r-- 1 andersen andersen  1761280 Apr 14 17:47 /tmp/bbox.tar
        $ gzip /tmp/busybox.tar
        $ ls -la /tmp/busybox*
        -rw-rw-r-- 1 andersen andersen   554058 Apr 14 17:49 /tmp/bbox.tar.gz
```

### halt

halt
Halt the system.

### head

head [OPTION] [FILE]...
Print first 10 lines of each FILE to standard output. With more than one FILE, precede each with a header giving the file name. With no FILE, or when FILE is -, read standard input.
Options:

```
        -n NUM          Print first NUM lines instead of first 10
```

Example:

```
        $ head -n 2 /etc/passwd
        root:x:0:0:root:/root:/bin/bash
        daemon:x:1:1:daemon:/usr/sbin:/bin/sh
```

### hostname

hostname [OPTION] {hostname | **-F** FILE}
Get or set the hostname or DNS domain name. If a hostname is given (or FILE with the **-F** parameter), the host name will be set.
Options:

```
        -s              Short
        -i              Addresses for the hostname
        -d              DNS domain name
        -F, --file FILE Use the contents of FILE to specify the hostname
```

Example:

```
        $ hostname
        sage
```

### *id*

id [OPTIONS]... [USERNAME]
Print information for USERNAME or the current user
Options:

```
        -g      prints only the group ID
        -u      prints only the user ID
        -n      print a name instead of a number (with for -ug)
        -r      prints the real user ID instead of the effective ID (with -
ug)
```

Example:

```
        $ id
        uid=1000(andersen) gid=1000(andersen)
```

### *ifconfig*

ifconfig **[-a]** <interface> [<address>]
configure a network interface
Options:

```
        [[-]broadcast [<address>]]  [[-]pointopoint [<address>]]
        [netmask <address>]  [dstaddr <address>]
        [outfill <NN>] [keepalive <NN>]
        [hw ether <address>]  [metric <NN>]  [mtu <NN>]
        [[-]trailers]  [[-]arp]  [[-]allmulti]
        [multicast]  [[-]promisc]  [txqueuelen <NN>]  [[-]dynamic]
        [mem_start <NN>]  [io_addr <NN>]  [irq <NN>]
        [up|down] ...
```

### *insmod*

insmod [OPTION]... MODULE [symbol=value]...
Loads the specified kernel modules into the kernel.
Options:

```
        -f      Force module to load into the wrong kernel version.
        -k      Make module autoclean-able.
        -v      verbose output
        -L      Lock to prevent simultaneous loads of a module
        -x      do not export externs
```

### *ipchains*

This is for firewalling. (See *Firewalling with ipchains*)

### *ipmasqadm*

This is for ip masquerading. (See *Masqueradin*).

### *kill*

kill **[-signal]** process-id [process-id ...]

Send a signal (default is SIGTERM) to the specified process(es).
Options:

```
-l      List all signal names and numbers.
```

Example:

```
$ ps | grep apache
252 root     root     S [apache]
263 www-data www-data S [apache]
264 www-data www-data S [apache]
265 www-data www-data S [apache]
266 www-data www-data S [apache]
267 www-data www-data S [apache]
$ kill 252
```

### *killall*

killall [**-signal**] process-name [process-name ...]
Send a signal (default is SIGTERM) to the specified process(es).
Options:

```
-l      List all signal names and numbers.
```

Example:

```
$ killall apache
```

### *ln*

ln [OPTION] TARGET... LINK_NAME|DIRECTORY
Create a link named LINK_NAME or DIRECTORY to the specified TARGET
You may use '--' to indicate that all following arguments are non-options.
Options:

```
-s      make symbolic links instead of hard links
-f      remove existing destination files
-n      no dereference symlinks - treat like normal file
```

Example:

```
$ ln -s BusyBox /tmp/ls
$ ls -l /tmp/ls
lrwxrwxrwx 1 root     root          7 Apr 12 18:39 ls -> BusyBox*
```

### *logger*

logger [OPTION]... [MESSAGE]
Write MESSAGE to the system log. If MESSAGE is omitted, log stdin.
Options:

```
-s      Log to stderr as well as the system log.
-t      Log using the specified tag (defaults to user name).
-p      Enter the message with the specified priority.
        This may be numerical or a ``facility.level'' pair.
```

Example:

```
$ logger "hello"
```

### *ls*

ls [-1AacCdeFilnpLRrSsTtuvwxXhk] [filenames...]
List directory contents
Options:

```
-1      list files in a single column
```

```
        -A      do not list implied . and ..
        -a      do not hide entries starting with .
        -C      list entries by columns
        -c      with -l: show ctime
        -d      list directory entries instead of contents
        -e      list both full date and full time
        -F      append indicator (one of */=@|) to entries
        -i      list the i-node for each file
        -l      use a long listing format
        -n      list numeric UIDs and GIDs instead of names
        -p      append indicator (one of /=@|) to entries
        -L      list entries pointed to by symbolic links
        -R      list subdirectories recursively
        -r      sort the listing in reverse order
        -S      sort the listing by file size
        -s      list the size of each file, in blocks
        -T NUM  assume Tabstop every NUM columns
        -t      with -l: show modification time
        -u      with -l: show access time
        -v      sort the listing by version
        -w NUM  assume the terminal is NUM columns wide
        -x      list entries by lines instead of by columns
        -X      sort the listing by extension
        -h      print sizes in human readable format (e.g., 1K 243M 2G )
        -k      print sizes in kilobytes(default)
```

### *lsmod*

lsmod<br>
List the currently loaded kernel modules.

### *makedevs*

makedevs NAME TYPE MAJOR MINOR FIRST LAST [s]
Creates a range of block or character special files
TYPEs include:

```
        b:      Make a block (buffered) device.
        c or u: Make a character (un-buffered) device.
        p:      Make a named pipe. MAJOR and MINOR are.
```

FIRST specifies the number appended to NAME to create the first device. LAST specifies the number of the last item that should be created. If 's' is the last argument, the base device is created as well.
For example:

```
        makedevs /dev/ttyS c 4 66 2 63   ->  ttyS2-ttyS63
        makedevs /dev/hda b 3 0 0 8 s    ->  hda,hda1-hda8
```

Example:

```
        $ makedevs /dev/ttyS c 4 66 2 63
        [creates ttyS2-ttyS63]
        $ makedevs /dev/hda b 3 0 0 8 s
        [creates hda,hda1-hda8]
```

### *md5sum*

md5sum [OPTION] [FILE]... or: md5sum [OPTION] **-c** [FILE]
Print or check MD5 checksums.

Options: With no FILE, or when FILE is -, read standard input.

```
        -b      read files in binary mode
        -c      check MD5 sums against given list
        -t      read files in text mode (default)
        -g      read a string
```

The following two options are useful only when verifying checksums:

```
        -s      don't output anything, status code shows success
        -w      warn about improperly formated MD5 checksum lines
```

Example:

```
        $ md5sum < busybox
        6fd11e98b98a58f64ff3398d7b324003
        $ md5sum busybox
        6fd11e98b98a58f64ff3398d7b324003  busybox
        $ md5sum -c -
        6fd11e98b98a58f64ff3398d7b324003  busybox
        busybox: OK
        ^D
```

### *menu*

menu

Show the Coyote Linux configuration menu.

### *mkdir*

mkdir [OPTION] DIRECTORY...

Create the `DIRECTORY(ies)` if they do not already exist

Options:

```
        -m      set permission mode (as in chmod), not rwxrwxrwx - umask
        -p      no error if existing, make parent directories as needed
```

Example:

```
        $ mkdir /tmp/foo
        $ mkdir /tmp/foo
        /tmp/foo: File exists
        $ mkdir /tmp/foo/bar/baz
        /tmp/foo/bar/baz: No such file or directory
        $ mkdir -p /tmp/foo/bar/baz
```

### *mknod*

mknod [OPTIONS] NAME TYPE MAJOR MINOR

Create a special file (block, character, or pipe).

Options:

```
        -m      create the special file using the specified mode (default
a=rw)
```

TYPEs include:

```
        b:      Make a block (buffered) device.
        c or u: Make a character (un-buffered) device.
        p:      Make a named pipe. MAJOR and MINOR are.
```

Example:

```
        $ mknod /dev/fd0 b 2 0
        $ mknod -m 644 /tmp/pipe p
```

### *more*

more [FILE ...]
More is a filter for viewing FILE one screenful at a time.
Example:

```
$ dmesg | more
```

### *mount*

mount [flags] DEVICE NODE [**-o** options,more-options]
Mount a filesystem
Flags:

```
-a:             Mount all filesystems in fstab.
-f:             "Fake" Add entry to mount table but don't mount it.
-n:             Don't write a mount table entry.
-o option:      One of many filesystem options, listed below.
-r:             Mount the filesystem read-only.
-t fs-type:     Specify the filesystem type.
-w:             Mount for reading and writing (default).
```

Options for use with the ``**-o**'' flag:

```
async/sync:     Writes are asynchronous / synchronous.
atime/noatime:  Enable / disable updates to inode access times.
dev/nodev:      Allow use of special device files / disallow them.
exec/noexec:    Allow use of executable files / disallow them.
loop:           Mounts a file via loop device.
suid/nosuid:    Allow set-user-id-root programs / disallow them.
remount:        Re-mount a mounted filesystem, changing its flags.
ro/rw:          Mount for read-only / read-write.
bind:           Use the linux 2.4.x "bind" feature.
```

There are EVEN MORE flags that are specific to each filesystem. You'll have to see the written documentation for those filesystems.
Example:

```
$ mount
/dev/hda3 on / type minix (rw)
proc on /proc type proc (rw)
devpts on /dev/pts type devpts (rw)
$ mount /dev/fd0 /mnt -t msdos -o ro
$ mount /tmp/diskimage /opt -t ext2 -o loop
```

### *mv*

mv SOURCE DEST or: mv SOURCE... DIRECTORY
Rename SOURCE to DEST, or move `SOURCE(s)` to DIRECTORY.
Example:

```
$ mv /tmp/foo /bin/bar
```

### *netstat*

Network statistics.

### *nslookup*

nslookup [HOST] [SERVER]
Queries the nameserver for the IP address of the given HOST optionally using a specified DNS server

Example:

```
$ nslookup localhost
Server:     default
Address:    default

Name:       debian
Address:    127.0.0.1
```

### *pidof*

pidof process-name [process-name ...]
Lists the PIDs of all processes with names that match the names on the command line
Example:

```
$ pidof init
1
```

### *ping*

ping [OPTION]... host
Send ICMP ECHO_REQUEST packets to network hosts.
Options:

```
-c COUNT        Send only COUNT pings.
-s SIZE         Send SIZE data bytes in packets (default=56).
-q              Quiet mode, only displays output at start
                and when finished.
```

Example:

```
$ ping localhost
PING slag (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=255 time=20.1 ms

--- debian ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 20.1/20.1/20.1 ms
```

### *poweroff*

poweroff
Halt the system and request that the kernel shut off the power.

### *printenv*

printenv
Prints the current environment or runs a program after setting up the specified environment.

### *printf*

printf FORMAT [ARGUMENT...]
Formats and prints `ARGUMENT(s)` according to FORMAT, Where FORMAT controls the output exactly as in C printf.
Example:

```
$ printf "Val=%d\n" 5
Val=5
```

### *ps*

ps

Report process status

This version of ps accepts no options.

Example:

```
$ ps
  PID  Uid      Gid State Command
    1 root     root     S init
    2 root     root     S [kflushd]
    3 root     root     S [kupdate]
    4 root     root     S [kpiod]
    5 root     root     S [kswapd]
  742 andersen andersen S [bash]
  743 andersen andersen S -bash
  745 root     root     S [getty]
 2990 andersen andersen R ps
```

### *pwd*

pwd

Print the full filename of the current working directory.

Example:

```
$ pwd
/root
```

### *rdate*

rdate [OPTION] HOST

Get and possibly set the system date and time from a remote HOST.

Options:

```
-s      Set the system date and time (default).
-p      Print the date and time.
```

### *reboot*

reboot

Reboot the system.

### *reset*

reset

Resets the screen.

### *rm*

rm [OPTION]... FILE...

Remove (unlink) the FILE(s). You may use '--' to indicate that all following arguments are non-options.

Options:

```
-i          always prompt before removing each destination
    -f              remove existing destinations, never prompt
-r or -R    remove the contents of directories recursively
```

Example:

```
$ rm -rf /tmp/foo
```

### *rmdir*

rmdir [OPTION]... DIRECTORY...
Remove the DIRECTORY(ies), if they are empty.
Example:

```
        # rmdir /tmp/foo
```

### *rmmod*

rmmod [OPTION]... [MODULE]...
Unloads the specified kernel modules from the kernel.
Options:

```
        -a      Try to remove all unused kernel modules.
```

Example:

```
        $ rmmod tulip
```

### *route*

route [{add|del|flush}]
Edit the kernel's routing tables

### *sed*

sed **[-nef]** pattern [files...]
Options:

```
        -n               suppress automatic printing of pattern space
        -e script        add the script to the commands to be executed
        -f scriptfile    add the contents of script-file to the commands to be
executed
```

If no **-e** or **-f** is given, the first non-option argument is taken as the sed script to interpret. All remaining arguments are names of input files; if no input files are specified, then the standard input is read.
Example:

```
        $ echo "foo" | sed -e 's/f[a-zA-Z]o/bar/g'
        bar
```

### *showcfg*

showcfg
Show the running network configuration.

### *sleep*

sleep N
Pause for N seconds.
Example:

```
        $ sleep 2
        [2 second delay results]
```

### *sort*

sort **[-nru]** [FILE]...
Sorts lines of text in the specified files
Options:

```
        -u      suppress duplicate lines
        -r      sort in reverse order
        -n      sort numerics
```

Example:

```
        $ echo -e "e\nf\nb\nd\nc\na" | sort
        a
        b
        c
        d
        e
        f
```

### *stty*

stty [**-a**|g] [**-F** DEVICE] [SETTING]...

Without arguments, prints baud rate, line discipline, and deviations from stty sane.

Options:

```
        -F DEVICE       open device instead of stdin
        -a              print all current settings in human-readable form
        -g              print in stty-readable form
        [SETTING]       see manpage
```

### *sync*

sync

Write all buffered filesystem blocks to disk.

### *syslogd*

syslogd [OPTION]...

Linux system and kernel logging utility. Note that this version of syslogd ignores /etc/syslog.conf.

Options:

```
        -m NUM          Interval between MARK lines (default=20min, 0=off)
        -n              Run as a foreground process
        -O FILE         Use an alternate log file (default=/var/log/messages)
        -R HOST[:PORT]  Log to IP or hostname on PORT (default PORT=514/UDP)
        -L              Log locally and via network logging (default is
network only)
```

Example:

```
        $ syslogd -R masterlog:514
        $ syslogd -R 192.168.1.1:601
```

### *tail*

tail [OPTION]... [FILE]...

Print last 10 lines of each FILE to standard output. With more than one FILE, precede each with a header giving the file name. With no FILE, or when FILE is -, read standard input.

Options:

```
        -c N[kbm]       output the last N bytes
        -n N[kbm]       print last N lines instead of last 10
        -f              output data as the file grows
        -q              never output headers giving file names
        -s SEC          wait SEC seconds between reads with -f
        -v              always output headers giving file names
```

If the first character of N (bytes or lines) is a '+', output begins with the Nth item from the start of each file, otherwise, print the last N items in the file. N bytes may be suffixed by k (x1024), b (x512), or m (1024^2).
Example:

```
        $ tail -n 1 /etc/resolv.conf
        nameserver 10.0.0.1
```

### *tar*

tar **-**[cxtvO] [**--exclude** FILE] [**-X** FILE][**-f** TARFILE] [**-C** DIR] [FILE(s)] ...
Create, extract, or list files from a tar file.
Options:

```
        c               create
        x               extract
        t               list
```

File selection:

```
        f               name of TARFILE or "-" for stdin
        O               extract to stdout
        exclude         file to exclude
        X               file with names to exclude
        C               change to directory DIR before operation
        v               verbosely list files processed
```

Example:

```
        $ zcat /tmp/tarball.tar.gz | tar -xf -
        $ tar -cf /tmp/tarball.tar /usr/local
```

### *telnet*

telnet HOST [PORT]
Telnet is used to establish interactive communication with another computer over a network using the TELNET protocol.

### *test*

test EXPRESSION or [ EXPRESSION ]
Checks file types and compares values returning an exit code determined by the value of EXPRESSION.
Example:

```
        $ test 1 -eq 2
        $ echo $?
        1
        $ test 1 -eq 1
        $ echo $?
        0
        $ [ -d /etc ]
        $ echo $?
        0
        $ [ -d /junk ]
        $ echo $?
        1
```

### *time*

time [OPTION]... COMMAND [ARGS...]
Runs the program COMMAND with arguments ARGS.

### *touch*

touch [**-c**] FILE [FILE ...]

Update the last-modified date on the given FILE[s].

Options:

```
        -c      Do not create any files
```

Example:

```
        $ ls -l /tmp/foo
        /bin/ls: /tmp/foo: No such file or directory
        $ touch /tmp/foo
        $ ls -l /tmp/foo
        -rw-rw-r--    1 andersen andersen        0 Apr 15 01:11 /tmp/foo
```

### *tr*

tr [**-cds**] STRING1 [STRING2]

Translate, squeeze, and/or delete characters from standard input, writing to standard output.

Options:

```
        -c      take complement of STRING1
        -d      delete input characters coded STRING1
        -s      squeeze multiple output characters of STRING2 into one
character
```

Example:

```
        $ echo "gdkkn vnqkc" | tr [a-y] [b-z]
        hello world
```

### *traceroute*

traceroute [**-dnrv**] [**-m** max_ttl] [**-p** port#] [**-q** nqueries]
[**-s** src_addr] [**-t** tos] [**-w** wait] host [data
size]

trace the route ip packets follow going to ``host'' Options:

```
        -d      set SO_DEBUG options to socket
        -n      Print hop addresses numerically rather than symbolically
        -r      Bypass the normal routing tables and send directly to a host
        -v      Verbose output
        -m max_ttl      Set the max time-to-live (max number of hops)
        -p port#        Set the base UDP port number used in probes
                (default is 33434)
        -q nqueries     Set the number of probes per ``ttl'' to nqueries
                (default is 3)
        -s src_addr     Use the following IP address as the source address
        -t tos  Set the type-of-service in probe packets to the following
value
                (default 0)
        -w wait Set the time (in seconds) to wait for a response to a probe
                (default 3 sec.).
```

### *true*

true

Return an exit code of TRUE (0).

Example:

```
        $ true
        $ echo $?
```

```
0
```

### tty

tty

Print the file name of the terminal connected to standard input.

Options:

```
-s      print nothing, only return an exit status
```

Example:

```
$ tty
/dev/tty2
```

### umount

umount [flags] FILESYSTEM|DIRECTORY

Unmount file systems

Flags:

```
-a      Unmount all file systems in /etc/mtab
-n      Don't erase /etc/mtab entries
-r      Try to remount devices as read-only if mount is busy
-f      Force umount (i.e., unreachable NFS server)
-l      Do not free loop device (if a loop device has been used)
```

Example:

```
$ umount /dev/hdc1
```

### uname

uname [OPTION]...

Print certain system information. With no OPTION, same as **-s**.

Options:

```
-a      print all information
-m      the machine (hardware) type
-n      print the machine's network node hostname
-r      print the operating system release
-s      print the operating system name
-p      print the host processor type
-v      print the operating system version
```

Example:

```
$ uname -a
Linux debian 2.2.15pre13 #5 Tue Mar 14 16:03:50 MST 2000 i686 unknown
```

### uniq

uniq [OPTION]... [INPUT [OUTPUT]]

Discard all but one of successive identical lines from INPUT (or standard input), writing to OUTPUT (or standard output).

Options:

```
-c      prefix lines by the number of occurrences
-d      only print duplicate lines
-u      only print unique lines
```

Example:

```
$ echo -e "a\na\nb\nc\nc\na" | sort | uniq
a
b
c
```

### *update*

update [options]

Periodically flushes filesystem buffers.

Options:

```
-S      force use of sync(2) instead of flushing
-s SECS call sync this often (default 30)
-f SECS flush some buffers this often (default 5)
```

### *uptime*

uptime

Display the time since the last boot.

Example:

```
$ uptime
  1:55pm  up  2:30, load average: 0.09, 0.04, 0.00
```

### *wc*

wc [OPTION]... [FILE]...

Print line, word, and byte counts for each FILE, and a total line if more than one FILE is specified. With no FILE, read standard input.

Options:

```
-c      print the byte counts
-l      print the newline counts
-L      print the length of the longest line
-w      print the word counts
```

Example:

```
$ wc /etc/passwd
     31      46    1365 /etc/passwd
```

### *which*

which [COMMAND ...]

Locates a COMMAND.

Example:

```
$ which login
/bin/login
```

### *whoami*

whoami

Prints the user name associated with the current effective user id.

### *yes*

yes [OPTION]... [STRING]...

Repeatedly outputs a line with all specified STRING(s), or 'y'.

### *zcat*

zcat FILE

Uncompress to stdout.

## Linux Commands for DOS/Windows Users

| DOS/Windows Command | Linux Command |
|---|---|
| arp | arp |
| cls | clear |
| copy | cp |
| del | rm |
| dir | ls |
| hostname | hostname |
| mkdir | mkdir |
| ping | ping |
| rmdir | rm -r |
| tracert | traceroute |
| xcopy | cp -r |

## Appendix A: Supported Network Cards

This list is arranged by the module name required to run the network card.

### *3c501*

3com 3c501

### *3c503*

3Com Etherlink II 3c503 10BaseT ISA

### *3c505*

3com Etherlink Plus 3c505

### *3c507*

3com Etherlink16 3c507

### *3c509*

Etherlink III (3c509 and 3c579)

### *3c59x*

Fast Etherlink and Boomerang Etherlink XL Series
3Com 3C450 HomeConnect 10/100 PCI
3com 3c592
3com 3C592-Combo
3com 3C592-TPO
3com 3c595
3com 3c597
3Com 3C905B
3Com 3C905B-Combo
3Com 3C905C
3Com Etherlink III (3c590)
3Com 3C980B-TX
3com 3c900
3Com 3C900B-Combo
3com 3C900B-FL (ST)
3com 3C900B-TPO
3Com 3CSOHO100-TX

### *ac3200*

Ansel Communications EISA ethernet adaptor

### *apricot*

Apricot Xen-II On Board Ethernet

### *arcnet*

ARCnet All (ALPHA)
Fujitsu FMV-181A
Fujitsu FMV-182A
Fujitsu FMV-183A
Fujitsu FMV-184A

### *at1700*

Allied Telesis AT1700 (ALPHA)

### *atp*

AT-Lan-Tec/Realtek RTL8002 (chipset)
AT-Lan-Tec/Realtek RTL8012 (chipset)

### *de4x5*

AOpen AON-315
DEC DE-425 (TP BNC EISA)
DEC DE-434 (TP PCI)
DEC DE-435 (TP BNC AUI PCI)
DEC DE-450 (TP BNC AUI PCI)
DEC DE-500 10/100
DIGITAL DC21x4x DECchip
Kingston DEC 21x4x based cards
SMC EtherPower 10B-T PCI

### *de600*

D-Link DE-600

### *de620*

D-Link DE-620

### *depca*

DEC DEPCA
DEC DE-100
DEC DE-101
DEC DE-200 Turbo
DEC DE-201 Turbo
DEC DE-202 Turbo (TP BNC)
DEC DE-210
DEC DE-422 (EISA)
DEC EtherWORKS

### *e2100*

Cabletron E2100

### *eepro*

Intel EtherExpress Pro/10
eepro100
Intel EtherExpress Pro/100B

Intel Other i82557 based cards

### *eexpress*

Intel EtherExpress 16

### *ewrk3*

DEC DE-203 Turbo (BNC)
DEC DE-204 Turbo (TP)
DEC DE-205 Turbo (TP BNC)
DEC EtherWORKS III

### *hp*

Hewlett Packard PC-Lan

### *hp100*

Compex Readylink ENET100-VG4
Compex FreedomLine 100/VG (ISA, EISA, PCI)
Hewlett Packard 100VG-AnyLan
Hewlett Packard 27248B (Cascade)
Hewlett Packard J2577 (Cascade)
Hewlett Packard J2577 (REVA Cascade)
Hewlett Packard J2573 (Cascade)
Hewlett Packard J2573 (REVA Cascade)
Hewlett Packard J2585
Hewlett Packard J2585AB
Hewlett Packard J2970
Hewlett Packard J2973

### *hp-plus*

Hewlett Packard PC-Lan Plus

### *lance*

Allied Telesis AT1500
AMD Lance PCnet-ISA
AMD Lance PCnet-ISA+
AMD Lance PCnet-PCI II
AMD 79C960 based cards (BOCA, Kingston, Linksys, HP)
Hewlett Packard J2405A
NE1500

### *lance32*

AMD Lance Pcnet-PCI
AMD Lance Pcnet-32
AMD Lance Pcnet-Fast

### *ne*

AOpen AON-101
Compex Enet16/V
Compex Readylink 2000
D-Link DE-220PCT Isa
Genius KE2000
Kingston KNW20TX EthRx
Kingston KNW20BT EthRx
KTI ET32P2
Linksys LNE2000
NetGear EA201c ISA
NE1000
NE2000
Realtek 8029
Realtek RTL8019
Surecom NE34
VIA 82C926 Amazon
Winbond 89C940
ne2k-pci
Amcom E450
AOpen AON-201
AOpen/Acer 90.80316.A12
Chipset: 8390
D-Link DE 528CT PCI NIC Card, RealTek Chipset
Encore ESL-816V/816V-T
Genius GE2000III SE
Genius GH4050 Hub Card
Genius GH4050C Hub Card
Genius GE2500III SE PCI
Linksys LNEPCI2 (10 Mbps ethernet card with a NE2000 chipset)
SOHO-PCI

### *ni52*

NI5210

### *ni65*

NI6510

### *rtl8139*

Kingston KNE120TX 10/100TX PCI
Realtek RTL8129/8139 Fast Ethernet
D-Link DFE-530TX+

### *seeq8005*

SEEQ 8005 based cards

### *sk_g16*

Schneider & Koch G16

### *smc-ultra*

SMC Ultra
SMC EtherEZ
SMC 8216

### *smc-ultra32*

SMC Ultra32 EISA
SMC 83c790 chips

### *smc9194*

SMC 9000 Series

### *tlan*

Compaq Netelligent 10 (PCI)
Compaq Netelligent 10/100 (PCI)
Compaq Integrated NetFlex-3/P (PCI)
Compaq NetFlex-3/P (PCI)
Compaq ProLiant Netelligent 10/100 (PCI)
Compaq Dual Port Netelligent 10/100 (PCI)
Compaq Deskpro 4000 5233MMX (PCI)
Compaq Netelligent 10 T2 (PCI)
Compaq Netelligent 10/100 TX (PCI)
Compaq Netelligent 10/100 TX UDP (PCI)
Compaq Netelligent 10/100 TX w/ embedded UTP (PCI)
Compaq Integrated Netelligent 10/100 TX (PCI)
Compaq Integrated NetFlex-3/P (PCI)
Compaq Dual Port Netelligent 10/100 TX (PCI)
Olicom OC-2325
Olicom OC-2183
Olicom OC-2326

### *tulip*

Accton EtherDuo PCI
Accton EN1207 (all three types)
Adaptec ANA6901/C
Adaptec ANA6911/TX
Allied Telesis LA100PCI-T
C-Net Pro120(C) (MX98715AEC chipset)
C-NET CNE-935
Cogent EM100
Cogent EM110
Cogent EM400
Cogent EM960
Cogent EM964 Quartet
Danpex EN-9400P3
DEC EtherWORKS 10 (PCI)
DEC EtherWORKS 10/100 (PCI)
DEC QSILVER
DEC chips 21040/21041/21140
DEC chips 21140A/21142
D-Link DE-530CT
D-Link DFE500-Tx
Kingston EtherX KNE100TX
Kingston EtherX KNT40T
Linksys EtherPCI
Linksys LNE100tx
LinkSys NC100 10/100 PCI
NetGear FA-310TX 10/100 PCI
SMC EtherPower 10 (PCI)
SMC EtherPower 10/100 (PCI)
SMC EtherPower Combo
Surecom EP-320X
Thomas Conrad TC 5048
Zynx ZX312 EtherAction
Zynx ZX314
Zynx ZX315 EtherArray
Zynx ZX342/344/345/346/348/351

### *[via-rhine]*

D-Link DFE-530TX
D-Link DFE-538TX
VIA 86c100A Rhine

### *[wd]*

Pure Data PDI8023-8
Pure Data PDUC8023
Pure Data PDI8023-16
SMC 8013WD
Western Digital WD8003
Western Digital WD8013

## Appendix B: Placing Coyote Linux on a CD

This is not really a supported function of Coyote Linux, but since many people desire to do this here are the instructions.

You will need to have a good Coyote Linux floppy already made and configured to your taste as it will not be able to make changes to you configuration once it is burnt to a CD, obviously.

Mount your Coyote floppy on a non-Coyote Linux machine with a CD burner and create yourself a working directory:

```
not-coyote# cd
not-coyote# mkdir coyote-cdrom
not-coyote# cd coyote-cdrom
not-coyote# cp /mnt/floppy/* .
not-coyote# mkdir root
```

Now, we need to unpack the root module so that we can work with it:

```
not-coyote# cd root
not-coyote# tar -xzvf ../root.tgz
```

Next, we need to edit linuxrc and change the following entries and then save the file:

| Old line | New line |
|---|---|
| qt mkdir $MNT | #qt mkdir $MNT |
| qt mount -o ro -t $FSTYPE /dev/$DEVICE $MNT | #qt mount -o ro -t $FSTYPE /dev/$DEVICE $MNT |
| qt umount $MNT | #qt umount $MNT |

Then, edit the syslinux.cfg and change the boot device from a your floppy device to your CDROM device. The entry should look something like this boot=/dev/fd0u1680 and it needs changed to something like this /dev/sda.

Now, we need to move all your package files to a new directory. Look in packages and root.packages, all these files need copied.

```
not-coyote# cd var/lib/lrppkg
not-coyote# mkdir mnt
not-coyote# cd mnt
not-coyote# ls ../../../../../*.tgz
not-coyote# mv ../../../../../*.tgz .
```

Now, we need to pack up our root package:

```
not-coyote# cd ../../../..
not-coyote# tar -czvf ../root-new.tgz *
```

Check that root.tgz and root-new.tgz have the same permissions (chmod root-new.tgz if it doesn't match) and then replace root.tgz with root-new.tgz:

```
not-coyote# cd ..
not-coyote# ls -l  root*.tgz
-rw-r--r--    1 root     root      1064960 Apr  5 18:04 root.tgz
-rw-r--r--    1 root     root      1064960 Apr  5 18:04 root-new.tgz
not-coyote# rm root.tgz
not-coyote# mv root-new.tgz root.tgz
```

Finally, we prepare to create the CD:

```
not-coyote# cd
not-coyote# mkdir cdimg
not-coyote# mkdosfs -C 288.img 2880
not-coyote# mount -t msdos -o loop 288.img cdimg
not-coyote# cp ~/coyote-cdrom/* ./cdimg
not-coyote# syslinux 288.img
not-coyote# umount cdimg
not-coyote# mv 288.img cdimg
not-coyote# mkisofs -b 288.img -c boot-catalog -o /tmp/coyote.iso cdimg
not-coyote# cdrecord speed=4 dev=0,0,0 /tmp/coyote.iso
```

I have not personally tested this as I have no use for this type of setup right now, so I apologize for any errors. For the original document on this type of setup try this link:
(http://www.linuxrouter.org/listarch/linux-router/2001-03-01/frm00051.html)

## Change Log

| | |
|---|---|
| 4/6/2003 | Initial creation. |

## Translations

This document was originally written in English, but the following translations are available. Please note that the translations may not be completely up to date with the original document.