13.2. Interface Configuration Files

Interface configuration files control the software interfaces for individual network devices. As the system boots, it uses these files to determine what interfaces to bring up and how to configure them. These files are usually named **ifcfg-*<name>***, where ***<name>*** refers to the name of the device that the configuration file controls.

13.2.1. Ethernet Interfaces

One of the most common interface files is **ifcfg-eth0**, which controls the first Ethernet *network interface card* or *NIC* in the system. In a system with multiple NICs, there are multiple **ifcfg-eth*<X>*** files (where ***<X>*** is a unique number corresponding to a specific interface). Because each device has its own configuration file, an administrator can control how each interface functions individually.

The following is a sample **ifcfg-eth0** file for a system using a fixed IP address:

```
DEVICE=eth0
BOOTPROTO=none
ONBOOT=yes
NETWORK=10.0.1.0
NETMASK=255.255.255.0
IPADDR=10.0.1.27
USERCTL=no
```

The values required in an interface configuration file can change based on other values. For example, the **ifcfg-eth0** file for an interface using DHCP looks different because IP information is provided by the DHCP server:

```
DEVICE=eth0
BOOTPROTO=dhcp
ONBOOT=yes
```

The **Network Administration Tool** (**system-config-network**) is an easy way to make changes to the various network interface configuration files (refer toChapter 14, *Network Configuration* for detailed instructions on using this tool).

However, it is also possible to manually edit the configuration files for a given network interface.

Below is a listing of the configurable parameters in an Ethernet interface configuration file:

BOOTPROTO=*<protocol>*

where ***<protocol>*** is one of the following:

- **none** — No boot-time protocol should be used.
- **bootp** — The BOOTP protocol should be used.
- **dhcp** — The DHCP protocol should be used.

BROADCAST=*<address>*

where ***<address>*** is the broadcast address. This directive is deprecated, as the value is calculated automatically with **ifcalc**.

DEVICE=*<name>*

where *<name>* is the name of the physical device (except for dynamically-allocated PPP devices where it is the *logical name*).

DHCP_HOSTNAME

Use this option only if the DHCP server requires the client to specify a hostname before receiving an IP address.

DNS*{1,2}*=*<address>*

where *<address>* is a name server address to be placed in **/etc/resolv.conf** if the **PEERDNS** directive is set to **yes**.

ETHTOOL_OPTS=*<options>*

where *<options>* are any device-specific options supported by **ethtool**. For example, if you wanted to force 100Mb, full duplex:

```
ETHTOOL_OPTS="autoneg off speed 100 duplex full"
```

> **Note**
>
> Changing speed or duplex settings almost always requires disabling autonegotiation with the **autoneg off** option. This needs to be stated first, as the option entries are order-dependent.

GATEWAY=*<address>*

where ***<address>*** is the IP address of the network router or gateway device (if any).

HWADDR=*<MAC-address>*

where ***<MAC-address>*** is the hardware address of the Ethernet device in the form ***AA:BB:CC:DD:EE:FF***. This directive is useful for machines with multiple NICs to ensure that the interfaces are assigned the correct device names regardless of the configured load order for each NIC's module. This directive should **not** be used in conjunction with **MACADDR**.

IPADDR=*<address>*

where *<address>* is the IP address.

MACADDR=*<MAC-address>*

where ***<MAC-address>*** is the hardware address of the Ethernet device in the form ***AA:BB:CC:DD:EE:FF***. This directive is used to assign a MAC address to an interface, overriding the one assigned to the physical NIC. This directive should **not** be used in conjunction with **HWADDR**.

MASTER=*<bond-interface>*

where *<bond-interface>* is the channel bonding interface to which the Ethernet interface is linked.

This directive is used in conjunction with the **SLAVE** directive.

Refer to Section 13.2.3, "Channel Bonding Interfaces" for more information about channel bonding interfaces.

NETMASK=*<mask>*

where *<mask>* is the netmask value.

NETWORK=*<address>*

where *<address>* is the network address. This directive is deprecated, as the value is calculated automatically with **ifcalc**.

ONBOOT=*<answer>*

where *<answer>* is one of the following:

- **yes** — This device should be activated at boot-time.
- **no** — This device should not be activated at boot-time.

PEERDNS=*<answer>*

where *<answer>* is one of the following:

- **yes** — Modify **/etc/resolv.conf** if the DNS directive is set. If using DHCP, then **yes** is the default.
- **no** — Do not modify **/etc/resolv.conf**.

SLAVE=*<bond-interface>*

where *<bond-interface>* is one of the following:

- **yes** — This device is controlled by the channel bonding interface specified in the **MASTER** directive.
- **no** — This device is *not* controlled by the channel bonding interface specified in the **MASTER** directive.

This directive is used in conjunction with the **MASTER** directive.

Refer to Section 13.2.3, "Channel Bonding Interfaces" for more about channel bonding interfaces.

SRCADDR=*<address>*

where *<address>* is the specified source IP address for outgoing packets.

USERCTL=*<answer>*

where *<answer>* is one of the following:

- **yes** — Non-root users are allowed to control this device.
- **no** — Non-root users are not allowed to control this device.

13.2.2. IPsec Interfaces

The following example shows the **ifcfg** file for a network-to-network IPsec connection for LAN A. The unique name to identify the connection in this example is **ipsec1**, so the resulting file is named **/etc/sysconfig/network-scripts/ifcfg-ipsec1**.

```
TYPE=IPsec
ONBOOT=yes
IKE_METHOD=PSK
SRCNET=192.168.1.0/24
DSTNET=192.168.2.0/24
DST=X.X.X.X
```

In the example above, ***X.X.X.X*** is the publicly routable IP address of the destination IPsec router.

Below is a listing of the configurable parameters for an IPsec interface:

DST=*<address>*

where ***<address>*** is the IP address of the IPsec destination host or router. This is used for both host-to-host and network-to-network IPsec configurations.

DSTNET=*<network>*

where ***<network>*** is the network address of the IPsec destination network. This is only used for network-to-network IPsec configurations.

SRC=*<address>*

where ***<address>*** is the IP address of the IPsec source host or router. This setting is optional and is only used for host-to-host IPsec configurations.

SRCNET=*<network>*

where ***<network>*** is the network address of the IPsec source network. This is only used for network-to-network IPsec configurations.

TYPE=*<interface-type>*

where ***<interface-type>*** is **IPSEC**. Both applications are part of the **ipsec-tools** package.

If manual key encryption with IPsec is being used, refer to**/usr/share/doc/initscripts-***<version-number>***/sysconfig.txt** (replace***<version-number>*** with the version of the **initscripts** package installed) for configuration parameters.

The **racoon** IKEv1 key management daemon negotiates and configures a set of parameters for IPSec. It can use preshared keys, RSA signatures, or GSS-API. If **racoon** is used to automatically manage key encryption, the following options are required:

IKE_METHOD=*<encryption-method>*

where ***<encryption-method>*** is either **PSK**, **X509**, or **GSSAPI**. If **PSK** is specified, the **IKE_PSK** parameter must also be set. If **X509** is specified, the**IKE_CERTFILE** parameter must also be set.

IKE_PSK=*<shared-key>*

where ***<shared-key>*** is the shared, secret value for the PSK (preshared keys) method.

IKE_CERTFILE=*<cert-file>*

where ***<cert-file>*** is a valid **X.509** certificate file for the host.

IKE_PEER_CERTFILE=*<cert-file>*

where ***<cert-file>*** is a valid **X.509** certificate file for the *remote* host.

IKE_DNSSEC=*<answer>*

where ***<answer>*** is **yes**. The **racoon** daemon retrieves the remote host's**X.509** certificate via DNS. If a **IKE_PEER_CERTFILE** is specified, do *not*include this parameter.

For more information about the encryption algorithms available for IPsec, refer to the **setkey** man page. For more information about **racoon**, refer to the**racoon** and **racoon.conf** man pages.

13.2.3. Channel Bonding Interfaces

Red Hat Enterprise Linux allows administrators to bind multiple network interfaces together into a single channel using the **bonding** kernel module and a special network interface called a *channel bonding interface*. Channel bonding enables two or more network interfaces to act as one, simultaneously increasing the bandwidth and providing redundancy.

To create a channel bonding interface, create a file in the**/etc/sysconfig/network-scripts/** directory called **ifcfg-bond***<N>*, replacing***<N>*** with the number for the interface, such as **0**.

The contents of the file can be identical to whatever type of interface is getting bonded, such as an Ethernet interface. The only difference is that the **DEVICE=** directive must be **bond*<N>***, replacing ***<N>*** with the number for the interface.

The following is a sample channel bonding configuration file:

```
DEVICE=bond0
BOOTPROTO=none
ONBOOT=yes
NETWORK=10.0.1.0
NETMASK=255.255.255.0
IPADDR=10.0.1.27
USERCTL=no
```

After the channel bonding interface is created, the network interfaces to be bound together must be configured by adding the **MASTER=** and **SLAVE=**directives to their configuration files. The configuration files for each of the channel-bonded interfaces can be nearly identical.

For example, if two Ethernet interfaces are being channel bonded, both **eth0**and **eth1** may look like the following example:

```
DEVICE=eth<N>
BOOTPROTO=none
ONBOOT=yes
MASTER=bond0
SLAVE=yes
USERCTL=no
```

In this example, replace ***<N>*** with the numerical value for the interface.

For a channel bonding interface to be valid, the kernel module must be loaded. To ensure that the module is loaded when the channel bonding interface is brought up, add the following line to **/etc/modprobe.conf**:

```
alias bond<N> bonding
```

Replace ***<N>*** with the number of the interface, such as **0**. For each configured channel bonding interface, there must be a corresponding entry in**/etc/modprobe.conf**.

Once **/etc/modprobe.conf** is configured — and the channel bonding interface and network interfaces are configured — the **ifup** command can be used to bring up the channel bonding interface.

> **Important**
>
> Important aspects of the channel bonding interface are controlled through the kernel module. For more information about controlling the **bonding** modules, refer to Section 40.5.2, "The Channel Bonding Module".

13.2.4. Alias and Clone Files

Two lesser-used types of interface configuration files are *alias* and *clone* files.

Alias interface configuration files, which are used to bind multiple addresses to a single interface, use the **ifcfg-*<if-name>*:*<alias-value>*** naming scheme.

For example, an **ifcfg-eth0:0** file could be configured to specify**DEVICE=eth0:0** and a static IP address of

10.0.0.2, serving as an alias of an Ethernet interface already configured to receive its IP information via DHCP in**ifcfg-eth0**. Under this configuration, **eth0** is bound to a dynamic IP address, but the same physical network card can receive requests via the fixed, 10.0.0.2 IP address.

> **Caution**
>
> Alias interfaces do not support DHCP.

A clone interface configuration file should use the following naming convention: **ifcfg-*<if-name>*-*<clone-name>***. While an alias file allows multiple addresses for an existing interface, a clone file is used to specify additional options for an interface. For example, a standard DHCP Ethernet interface called **eth0**, may look similar to this:

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=dhcp
```

Since the default value for the **USERCTL** directive is **no** if it is not specified, users cannot bring this interface up and down. To give users the ability to control the interface, create a clone by copying **ifcfg-eth0** to **ifcfg-eth0-user** and add the following line to **ifcfg-eth0-user**:

```
USERCTL=yes
```

This way a user can bring up the **eth0** interface using the **/sbin/ifup eth0-user** command because the configuration options from **ifcfg-eth0** and **ifcfg-eth0-user** are combined. While this is a very basic example, this method can be used with a variety of options and interfaces.

The easiest way to create alias and clone interface configuration files is to use the graphical **Network Administration Tool**. For more information on using this tool, refer to Chapter 14, *Network Configuration*.

13.2.5. Dialup Interfaces

If you are connecting to the Internet via a dialup connection, a configuration file is necessary for the interface.

PPP interface files are named using the following format:

ifcfg-ppp*<X>*

where ***<X>*** is a unique number corresponding to a specific interface.

The PPP interface configuration file is created automatically when **wvdial**, the**Network Administration Tool** or **Kppp** is used to create a dialup account. It is also possible to create and edit this file manually.

The following is a typical **ifcfg-ppp0** file:

```
DEVICE=ppp0
NAME=test
WVDIALSECT=test
MODEMPORT=/dev/modem
LINESPEED=115200
PAPNAME=test
USERCTL=true
ONBOOT=no
PERSIST=no
DEFROUTE=yes
```

```
PEERDNS=yes
DEMAND=no
IDLETIMEOUT=600
```

Serial Line Internet Protocol (SLIP) is another dialup interface, although it is used less frequently. SLIP files have interface configuration file names such as **ifcfg-sl0**.

Other options that may be used in these files include:

DEFROUTE=*<answer>*

where *<answer>* is one of the following:

- **yes** — Set this interface as the default route.
- **no** — Do not set this interface as the default route.

DEMAND=*<answer>*

where *<answer>* is one of the following:

- **yes** — This interface allows **pppd** to initiate a connection when someone attempts to use it.
- **no** — A connection must be manually established for this interface.

IDLETIMEOUT=*<value>*

where *<value>* is the number of seconds of idle activity before the interface disconnects itself.

INITSTRING=*<string>*

where *<string>* is the initialization string passed to the modem device. This option is primarily used in conjunction with SLIP interfaces.

LINESPEED=*<value>*

where *<value>* is the baud rate of the device. Possible standard values include **57600**, **38400**, **19200**, and **9600**.

MODEMPORT=*<device>*

where *<device>* is the name of the serial device that is used to establish the connection for the interface.

MTU=*<value>*

where *<value>* is the *Maximum Transfer Unit (MTU)* setting for the interface. The MTU refers to the largest number of bytes of data a frame can carry, not counting its header information. In some dialup situations, setting this to a value of **576** results in fewer packets dropped and a slight improvement to the throughput for a connection.

NAME=*<name>*

where *<name>* is the reference to the title given to a collection of dialup connection configurations.

PAPNAME=*<name>*

where *<name>* is the username given during the *Password Authentication Protocol (PAP)* exchange that occurs to allow connections to a remote system.

PERSIST=*<answer>*

where *<answer>* is one of the following:

- **yes** — This interface should be kept active at all times, even if deactivated after a modem hang up.
- **no** — This interface should not be kept active at all times.

REMIP=*<address>*

where *<address>* is the IP address of the remote system. This is usually left unspecified.

WVDIALSECT=*<name>*

where *<name>* associates this interface with a dialer configuration in **/etc/wvdial.conf**. This file contains the phone number to be dialed and other important information for the interface.

13.2.6. Other Interfaces

Other common interface configuration files include the following:

ifcfg-lo

A local *loopback interface* is often used in testing, as well as being used in a variety of applications that require an IP address pointing back to the same system. Any data sent to the loopback device is immediately returned to the host's network layer.

> **Warning**
>
> The loopback interface script, **/etc/sysconfig/network-scripts/ifcfg-lo**, should never be edited manually. Doing so can prevent the system from operating correctly.

ifcfg-irlan0

An *infrared interface* allows information between devices, such as a laptop and a printer, to flow over an infrared link. This works in a similar way to an Ethernet device except that it commonly occurs over a peer-to-peer connection.

ifcfg-plip0

A *Parallel Line Interface Protocol (PLIP)* connection works much the same way as an Ethernet device, except that it utilizes a parallel port.

ifcfg-tr0

Token Ring topologies are not as common on *Local Area Networks* (*LANs*) as they once were, having been eclipsed by Ethernet.