WANPIPE

Multi-protocol WANPIPE Driver for Linux®

INSTALLATION MANUAL

SANGOMA TECHNOLOGIES

By Nenad Corbic

LIMITED USE WARRANTY

Sangoma Technologies

Markham, Ontario, Canada

Contents

Standards Used in This Manual

The table below shows text conventions used in this manual.

Table 1 - Text Conventions

Text	Description
[ ]	Text within square brackets represents keyboard keys. For example [Enter] or [p]. Combinations of keys held down together are indicated using the plus sign as in the example: [Alt+p].
< >	Text within pointy brackets represents any command or argument after a command. For example, <arg> represents an argument.
`Courier Font`	Text in this font indicates a directory name, a file name or a command string.
Bold	Bold text is used to draw attention to a new concept, or to highlight choices in description of a configuration.

The table below displays icons used to draw attention to items of note.

Table 2 - Icon Conventions

Icon	Description
ⓘ	Information, general interest note.
✪	Important, recommended procedure or configuration note.
⚠	Critical procedure or configuration note.

1. Introduction

The WANPIPE WAN Routing/API solution is comprised of Sangoma "S" series intelligent multi-protocol WAN T1/E1/ADSL adapters or cards, Linux kernel device drivers and a suite of configuration, operation and debugging utilities. All WANPIPE kernel device drivers are part of the standard Linux Kernel distribution.

Routing Solutions

With reference to the Open System Interconnection (OSI) protocol framework, the Sangoma “S” adapters along with the WANPIPE device drivers provide a physical and WAN data link layer to the Linux Kernel IP Routing Stack. The Linux IP Stack, using its powerful internal routing table, routes packets to and from the WANPIPE drivers. The full power of Linux stacks and applications are engaged over WANPIPE.

WANPIPE S-series adapters support data transfer rates up to 8Mbps. The majority of WAN protocols supported by WANPIPE are implemented in firmware and run on the card. As an intelligent adapter, it offloads demands on the system CPU and improves system stability. By adding a Sangoma WAN/ADSL component to the proven reliability of the Linux kernel, a powerful multi-T1/E1/ADSL router/firewall is created. Sangoma S-series cards support optional on-board T1/E1 CSU/DSU. In other words, a T1/E1 line can be connected directly to the card eliminating all external components found in a traditional routing solution.

The following WAN protocols are supported: ATM, Frame Relay, PPP, MULTILINK PPP and CHDLC.

API Solutions

WANPIPE supports a full socket based API stack that can be used to develop custom applications over WAN protocols.

Sangoma API protocols include: X25(API), Frame Relay (API), HDLC (API), BitStreaming (API), BiSync (API), SDLC (API) and SS7 (API).

Using the WAN API suite, a vast range of applications can be developed, such as: Credit card verification, Voice-over IP, Protocol/Line Data Scope, Satellite Comm and Legacy interconnect.

WANPIPE Software Package

This guide describes the steps to install and compile WANPIPE drivers. The first step is to get a copy of the software package release. It is available from two sources:

- Sangoma FTP download
- Sangoma CD-ROM

This section describes steps to copy installation files to a location on your machine.

Sangoma WANPIPE FTP Download

The URL for Sangoma FTP is: ftp://ftp.sangoma.com.

WANPIPE software is packaged in the following formats:

- Source tgz package

  FTP directory `[/linux/current_wanpipe]`

- RPM binary packages for Red Hat and other distributions

  FTP directory `[/linux/RPM]`

  (See Chapter 3)

- Linux Routing Project (LRP) binary package

  FTP directory `[/linux/LRP]`

  (See Chapter 4)

ⓘ

If you have problems downloading the drops, contact the webmaster.

If you have problems installing or getting the workbench to run, send an email to Sangoma support (instructions in the Debug Manual).

All downloads are provided under the terms and conditions of the Sangoma Warranty unless otherwise specified.

For information about older builds look in the `old_releases` folder.

Software Version

WANPIPE software is distributed in two release streams: **Stable** and **Beta**.

Our beta testers have tested the **Stable** stream in the Sangoma Lab and in the field for a minimum of one month. Changes in the Stable stream version releases reflect cosmetic updates and minor changes to configuration settings. No new features are added into the stable stream. Stable release file naming convention includes the driver utility name, version number and release number.

(Stable example: `wanpipe-2.2.5-1.tgz`)

The **Beta** stream has been fully tested in the Sangoma Lab and it is in the process of field testing at our customers' sites. The Beta stream usually contains major feature enhancements as well as any bug fixes found from previous releases. Beta release file naming convention includes the driver utility name, version, and beta label with release number and a version number.

(Beta example: `wanpipe-beta1-2.2.5.tgz`)

Sangoma CD-ROM

Another way to receive these drivers is on a CD-ROM from your sales representative. Follow these steps:

1. Mount the CD-ROM. For example, at the prompt enter:

   ```
   mount /dev/cdrom /mnt
   ```

2. Copy the `wanpipe[version_number].tgz` compressed file from the CD-ROM `linux/` directory into the root directory of your system. For example, at the prompt enter:

   ```
   cp /mnt/Unix-type\ systems/linux/wanpipe-X.Y.Z.tgz /tmp
   ```

2. Installing WANPIPE Source Package

The Setup installation script, located in `wanpipe temporary directory` (created on untaring the package), will guide you through the installation procedure. The installation utility will probe the kernel for previous WANPIPE installations and will offer appropriate installation options.

Setup can re-compile and install new WANPIPE kernel device driver modules based on the currently running Linux kernel, eliminating the need to recompile the kernel image manually.

Unpacking the Source Package

1. Log in as super user (root) and change your current directory to any temporary directory. For example:

   ```
   /root/tmp
   ```

   You can also use the 'su -' command to log in as root from any other user account.

2. Copy the WANPIPE distribution package (downloaded from Sangoma ftp site, or CD-ROM) to the temporary directory and un-tar it. For example:

   ```
   tar xvfz wanpipe-2.2.5-3.tgz
   ```

   The source distribution will create the `wanpipe/` directory.

Package Components

The table below describes the required **wanpipe Kernel drivers** in order of dependencies:

Table 3 - WANPIPE Kernel Drivers

File Name	Description
kernel/drivers/net/wan/ sdladrv.o	WANPIPE device hardware layer Module containing hardware functions used to detect and setup wanpipe adapters. i.e. PCI/ISA setup, memory test. Size: 20K
kernel/net/wanrouter/ wanrouter.o	WANPIPE kernel-user interface layer Module that sets up wanpipe /proc file system and handles configuration ioctl() Size: 16K
kernel/drivers/net/wan/ wanpipe_syncppp.o	WANPIPE Sync-PPP protocol Module containing complete PPP protocol implementation. Size: 12K
kernel/drivers/net/wan/ wanpipe.o	WANPIPE Multi-Protocol layer Module that implements all wanpipe WAN protocols. Frame Relay, PPP, CHDLC, ATM, ADSL ... Size: 700K-95K (i.e. 90K per protocol)
kernel/net/wanrouter/ af_wanpipe.o	WANPIPE API Socket layer Module that implements a wanpipe API "zero-loss" socket used to develop custom applications over WANPIPE protocols. Size: 16K

The table below describes the **required binaries** used to configure, operate and debug WANPIPE drivers.

Table 4 - Required Binaries

File Name	Description
/usr/sbin/wanrouter	WANPIPE start-up script Utility used to start/stop/restart wanpipe card
/usr/sbin/wanconfig	WANPIPE configuration utility Utility used to parse the text configuration file and

File Name	Description
	configure wanpipe kernel driver
/usr/sbin/wanpipemon	WANPIPE Debug Utility Utility used to obtain traces and statistics from wanpipe kernel drivers
/usr/sbin/sdladump	WANPIPE Hardware Memory Viewer Utility used to debug wanpipe firmware
/usr/sbin/cfgft1	FT1 CSU/DSU Configuration Utility Utility used to configure onboard CSU/DSU on S514-3/S508FT1 cards
/usr/sbin/wancfg	WANPIPE Configuration Utility Utility used to create the wanpipe configuration file

The table below describes **optional binaries** used with WANPIPE drivers.

Table 5 - Optional Binaries

File Name	Description
/usr/sbin/wpkbdmon	WANPIPE Keyboard Led Monitor Daemon used to display wanpipe driver statistics ((tx/rx/state) using keyboard leds. (OPTIONAL)
/usr/sbin/wpbackupd	WANPIPE Backup Daemon Daemon used to provide wanpipe backup functionality to a secondary link.(OPTIONAL)

The table below describes **WANPIPE directories** used to hold configuration and environment files.

Table 6 - WANPIPE Directories

Directory	Description
/usr/share/doc/Wanpipe	Wanpipe documentation files
/etc/wanpipe/wanrouter.rc	Environment configuration file File containing all wanpipe environment paths and constants for all wanpipe utilities.
/etc/wanpipe/interfaces/*	Interface configuration files

Directory	Description
	Directory containing the IP information for all wanpipe network interfaces.
/etc/wanpipe/firmware/*	Firmware modules Directory containing the firmware modules for all wanpipe protocols.
/etc/wanpipe/lib	Library files used by wanpipe utilities
/etc/wanpipe/samples	Configuration Sample files Directory containing wanpipe configuration sample files, in case wancfg utility cannot be used.
/etc/wanpipe/api	WAN API sample codes for API protocols Directory containing wanpipe API sample C files used to aid in custom API development.

Source Package Installation Prerequisites

Be certain the following is in order before beginning the installation process:

- Full Linux source code in the `/usr/src/linux` directory (if not, the directory may be specified during installation.
- The kernel source in `/usr/src/linux` must have the same configuration and version number as the currently running image.

ⓘ *The Setup installation utility will check for a name mismatch between the image and source. User will be prompted if a mismatch is found.*

- Development system installed: GNU C compiler
- BASH version 2 installed

  (Required by WANPIPE utilities. Default on most systems).

Installation Process

For custom Source package installations using optional parameters, see **Customizing WANPIPE Source Package Setup** later in this guide.

To start WANPIPE installation, follow these steps:

1. From the **wanpipe** directory, run the **Setup** installation program using the following command:

   ```
   ./Setup install
   ```

   The **WANPIPE INSTALLATION** screen appears.

   Press **[y]** and **[Enter]** to execute the step.

   This first step will:

   - Verify distribution integrity and fix file permissions
   - Check for required packages, BASH2 installation, ncurses library and GCC.

.✪ *Note: If the Setup fails the BASH or GCC check, installation will stop.*

   Once the checks are complete, press **[Enter]** to continue installation.

2. The **UPDATING WANPIPE DEVICE DRIVERS: LINUX KERNEL** screen appears.

Press **[y]** and **[Enter]** to execute the step.

This step will:

- Probe the Linux kernel for previous WANPIPE installation.
- If this is a new installation, the kernel will be clean of any WANPIPE drivers. Appropriate patches are copied to the kernel source.
- If the kernel already has WANPIPE installed, drivers are upgraded to the latest version.

Note: Upgrading drivers is always the recommended practice. This way the latest sources are always in the kernel tree.

Once the Linux drivers are upgraded, press **[Enter]** to continue installation.

3. The **WANPIPE KERNEL DRIVER COMPILATION** screen appears.

Press **[y]** and **[Enter]** to execute the step.

This option will attempt to compile currently installed WANPIPE device drivers located in the Linux source tree and install WANPIPE modules.

The user has two compilation options:

- **Default** compilation option will compile the standard WANPIPE protocols into the kernel device driver: Frame Relay, PPP, CHDLC and X25(API).
- **Custom** compilation option will offer a menu of all available WAN and API protocols. The user will be able to enable protocols individually or all at once, as desired.

After selecting the protocol compilation mode, a screen report lists all selected WAN protocols to be compiled into the WANPIPE kernel device driver.

Once the compile process is complete, press **[Enter]** to continue installation.

Note: After this step is complete, it is not necessary to recompile the Kernel image. The newly compiled WANPIPE modules can be loaded directly into *the existing Kernel image.*

Note: If Kernel driver compilation fails, call Sangoma Technical Support.

4. The **WANPIPE BOOTSTRAP CONFIGURATION** screen appears.

   Press **[y]** and **[Enter]** to start WANPIPE along with system boot and stop on system shutdown.

   Press **[n]** and **[Enter]** to disable WANPIPE startup on boot.

5. The **WANPIPE ENVIRONMENT CONFIGURATION** screen appears.

   Wanpipe utilities depend on the environment configuration file, named `/etc/wanpipe/wanrouter.rc.` It defines locations of WANPIPE lock, configuration and interface files.

   Press **[Enter]** to install the files, utilities and libraries.

   You will be prompted to define these paths. Use default values.

✪ *Note: Always use default values unless you are installing for a custom environment that needs special considerations.*

6. The **WANPIPE UTILITES SETUP** screen appears.

   Press **[y]** and **[Enter]** to execute the step.

   This option compiles all WANPIPE utilities to be written to the locations indicated previously.

This is the end of the installation process. For help with the results of this process, please follow directions in the section that starts on the next page following.

Results of the Installation Process

Depending on the outcome of the setup, follow instructions below. If the overall configuration is complete, proceed to **WANPIPE INSTALLATION: COMPLETE** instructions below left.

If the configuration is not complete, the utilities have been installed, but nothing will start until the modules have been compiled. In this case, proceed to **WANPIPE INSTALLATION: INCOMPLETE** instructions below right.

WANPIPE INSTALLATION: COMPLETE screen appears.	**WANPIPE INSTALLATION: INCOMPLETE** screen appears.
This message is a result of the steps taken in the installation process. This message appears to verify that all WANPIPE device driver modules and utilities have been compiled. Press **[Enter]** to install all files. The installation process is complete. The next step is to configure the WANPIPE driver and start the router. PLEASE SKIP THE NEXT SECTION AND PROCEED TO **WANPIPE CONFIGURATION MANUAL**.	The WANPIPE device drivers have not been compiled for one of two reasons: 1. The driver compilation step (number 3, above) has been purposely skipped. This is not an error. The user wishes to compile Kernel drivers manually. 2a. The compilation of Kernel drivers failed. 2b. The module dependencies failed or modules failed to load because of a conflict between the Kernel image version or configuration and the Kernel source version or configuration. For help in resolving the failure of modules to compile, call Sangoma Technical Support.

3. Customizing WANPIPE Source Package Setup

This chapter describes the ./Setup command, seven optional arguments for custom situations and six optional command extensions.

Commands described in this chapter are for advanced custom setup with multiple architectures.

Table 7 - ./Setup Command Arguments

Argument	Syntax	Description
	./Setup	Run ./Setup command without any arguments for a listing of command arguments and optional command extensions.
install	./Setup install	Install WANPIPE.
upgrade	./Setup upgrade	Upgrade WANPIPE kernel device drivers and utilities.

remove	./Setup remove	WANPIPE Source Package Un-Install
		Run the Setup script in temporary `wanpipe/` directory. The remove option will do the following:
		1. Remove all bootstrap WANPIPE scripts.
		2. Remove all utilities from the following directories:
		`/usr/sbin`
		`/etc/wanpipe`
		`/usr/share/doc/wanpipe/`
		After the 'Setup remove' command, it is safe to remove the whole temporary `wanpipe/` directory.
		For example: `rm -rf wanpipe/`

utility	/Setup utility	Re-compile all WANPIPE utilities in temporary `wanpipe/` directory after running `./Setup install`. Run `./Setup inst` to copy the new binaries into `/usr/sbin` directory.
drivers	./Setup drivers	Use this option to update kernel sources with the current version of the WANPIPE drivers and recompile and install WANPIPE kernel modules. The drivers argument will do the following: 1. Prompt the user to Install or Upgrade WANPIPE drivers into the Linux kernel source. 2. Prompt the user to re-compile all WANPIPE kernel drivers and install them as modules in the `current /lib/modules/<kernel-version>` directory.
buildrpm	./Setup buildrpm	Builds a custom RPM. All defaults will be used: 1. Configuration directory `/etc/wanpipe`.

Argument	Syntax	Description
		2. Start WANPIPE on boot up. 3. Recompile modules for the kernel source located in `/usr/src/linux` directory, or directory supplied with `--with-linux=<abs path>` extension. Refer to `README.rpmbuild` for more information.
helprpm	./Setup helprpm	RPM build help information.
<any key>	./Setup <any key>	Help.

⚠ *Note: In silent mode or buildrpm, <kernel version> is obtained from `/usr/src/linux/Makefile`. This ensures that the name of the source image is the same as the name of the current image.*

The following optional command extensions can be used with any of the command arguments in the table above.

Table 8 - ./Setup Optional Command Extensions

Extension	Syntax	Description
--silent	./Setup <arg> --silent	User interaction is suppressed. All default settings will be used: 1. Configuration directory `/etc/Wanpipe`. 2. Start WANPIPE on boot up. 3. Recompile modules for the kernel source in `/usr/src/linux`.
--builddir	./Setup <arg> --builddir=<tmp directory> Example: ./Setup utility –builddir=/tmp/wanpipe_build	Override build root and use a temporary directory. Setup will see the temporary directory as the system root. When customizing the WANPIPE package, we advise installing into a temporary virtual directory to prevent corruption of the existing environment. Once custom binary tgz or RPM distribution is complete, write the contents to the root file system. Note: The `<tmp directory>` must exist prior to running the command. The path name must be absolute. Default is set to the system root (/).

Extension	Syntax	Description
--with-linux	./Setup <arg> --with-linux=<directory> Example: `--with-linux=/usr/src/linux-2.4.4`	Build modules using a custom linux source directory. Default value is set to `/usr/src/linux`
--arch	./Setup <arg> --arch=<value> Example: `--arch=i686`	Build modules using a custom defined architecture. (i686 or i586). By default: (uname -m)
--split_rpms	./Setup <arg> --split_rpms Example: `./Setup buildrpm --split_rpms`	Splits the rpm build into util and modules packages. By doing so, user space utilities and kernel modules are split into two packages. Default: One rpm package containing both utilities and modules.
--protocol	./Setup <arg> --protocol=<PNAME> Example: `./Setup install --protocol=BSCSTRM`	Enable non-mainstream protocols. These options are also available in Custom module compilation mode during `./Setup install`. The mainstream protocol s compiled by default are: Frame Relay, PPP, CHDLC, X25, API stack Non-Mainstream Protocols: BSC : API for Multipoint Bisync BSCSTRM : API for Bisync Streaming BITSTRM : API for Bit Streaming SS7 : API for Level2 SS7 protocol EDU : API for Educational software POS : POS (S509/S515) SDLC : API for SDLC protocol

4. RPM Binary Installation

Sangoma WANPIPE FTP Download

Before installing the RPM, make sure that you have downloaded the appropriate package for your distribution, smp support and kernel. The URL for Sangoma FTP is: ftp://ftp.sangoma.com/linux/RPMS.

Automatic RPM Installation

There are files here for installation in `/linux/RPMS`. Use the following command syntax:

```
./Setup install <abs path>
```

Setup installation script will scan your system and suggest an RPM.

If the `<abs path>` is not supplied, Setup will assume that this process is part of a CD distribution. It will expect the RPM directory structure as on the CD or FTP site. For example:

```
RPMS/<distr>/<version>/
```

If the RPMS are not found locally, Setup can download the latest releases from the Sangoma FTP site.

Once the RPM is downloaded, Setup can proceed to install the RPM automatically with the `./Setup install` command.

Manual RPM Setup

Follow the steps below to set up RPM manually:

1. Determine the Linux Distribution (for example: RedHat 7.1).

   ```
   cat /etc/issue
   ```

2. Determine the Kernel image you are running (A result showing 2.4.2-2 is not an smp kernel. A result showing 2.4.2-2smp is an smp kernel).

   ```
   uname -r
   ```

3. Determine the processor type (for example: i686).

   ```
   uname -m
   ```

4. Based on the above information choose an RPM package using the following syntax:

   ```
   wanpipe-<version>.<linux distrib>.<kernel ver>.<arch>.rpm
   ```

   The example command string below uses the example results from steps 1, 2 and 3:

   ```
   ftp.sangoma.com/linux/RPM/redhat/7.1/i686/wanpipe-2.2.4-3.rh71up
   .k242-2.i686.rpm
   ```

5. Download the RPM package.

6. List the contents of the rpm

   ```
   rpm -qipl <wanpipe rpm name>
   ```

Note: Make sure that the `/lib/modules/<kernel name>` *directory matches your current kernel modules directory. That is to say the output of command* `'uname -r'` *must match the above* `<kernel name>` *directory.*

7. The rpm must be installed using the `--force` option, to override the old modules.

   ```
   rpm -i --force <wanpipe rpm name>
   ```

8. Read `'README-2.config'` for information on how to configure `wanpipe device.rpm.`

5. LRP Binary Installation

The LRP package contains:

- `wanpipe.lrp`: Wanpipe utilities and startup scripts
- `modules.lrp`: Wanpipe kernel modules

Follow the steps below:

1. Copy the LRP packages onto a DOS formatted disk.
2. Boot the system with the LRP boot up disk.

   Insert the disk containing WANPIPE LRP packages.

   Mount the disk using the following command:

   ```
   mount.boot /mnt
   ```
3. Change directory to `/mnt` and install each package. For example:

   ```
   lrpkg –i modules
   lrpkg –i wanpipe
   ```
4. Create the WANPIPE configuration files in `/etc/wanpipe` directory.

6. Linux Kernel Configuration/Compilation

Note: If the Setup installation script recompiled the modules, this step is not necessary.

Configuring the Linux Kernel

Change your current directory to `/usr/src/linux`:

make `menuconfig'`	(menu- driven configuration)
or make `xconfig'`	(if you are running X-Windows)

Note: Linux kernel configuration is a complicated process. Please read `/usr/src/linux/README` *for assistance.*

The Linux configuration script will pose a series of questions regarding system configuration. Most questions require single-letter answer:

[y] to enable the feature

[n] to disable it

[m] to implement the feature as a module

[h] brings up a help text (if available).

Note: It is important that you answer ***NO*** *to the following questions when configuring the kernel:*

- WAN Interfaces
    - Frame relay DLCI support (CONFIG_DLCI)

✪ *Note: It is important that you answer* ***YES or M*** *to the following questions when configuring the kernel:*

- Code Maturity Level Options
    - Prompt for development drivers (CONFIG_EXPERIMENTAL)
- Loadable Module Support
    - Enable loadable module support (CONFIG_MODULES)
    - Kernel daemon support (CONFIG_KMOD)
- General Setup
    - Networking support (CONFIG_NET)
    - PCI support (CONFIG_PCI).
- Networking Options
    - Packet Socket (CONFIG_PACKET)
    - Unix domain sockets (CONFIG_UNIX)
    - TCP/IP networking (CONFIG_INET)
    - IP: optimize as router not host (CONFIG_IP_ROUTER)
    - 802.1d Ethernet Bridging (CONFIG_BRIDGE)
    - WAN Router (CONFIG_WAN_ROUTER)
- Network Device Support
    - Network device support (CONFIG_NETDEVICES)
- Wan Interfaces
    - WAN Drivers (CONFIG_WAN_DRIVERS)
    - Vendor Sangoma (CONFIG_VENDOR_SANGOMA)

Below are some other recommended configurations:

Option	**Configuration**
WANPIPE Frame Relay support (CONFIG_WANPIPE_FR)	**[Y]** To connect to a frame relay network. **[N]** Frame relay support will not be included in the driver (saves about 16K of kernel memory).
WANPIPE Cisco HDLC support (CONFIG_WANPIPE_CHDLC)	**[Y]** To connect to a leased line using Cisco HDLC protocol (WANPIPE or API). **[N]** Cisco HDLC support will not be included in the driver.
WANPIPE PPP support (CONFIG_WANPIPE_PPP)	**[Y]** To connect to a leased line using PPP protocol (WANPIPE). **[N]** PPP support will not be included in the driver.
WANPIPE X25 support (CONFIG_WANPIPE_X25)	**[Y]** To connect to a leased line using X25 protocol (WANPIPE or API). **[N]** X25 support will not be included in the driver.
WANPIPE Multi-Port PPP support (CONFIG_WANPIPE_MULTPPP)	**[Y]** To connect to multiple leased line, using the kernel PPP protocol (WANPIPE). Since the protocol is implemented in the kernel PPP, protocol can run on both adapter ports. **[N]** Multi-Port PPP support will not be included in the driver.

Filesystems

/proc filesystem support (CONFIG_PROC_FS)

Re-build Linux Kernel

✪ *Note: Linux kernel configuration is a complicated process. Please read* `/usr/src/linux/README` *for assistance.*

If your machine is slow or has a small amount of memory, re-building Linux kernel may take several hours.

After successfully configuring the kernel use 'make menuconfig' to perform the following (from /usr/src/linux directory):

make dep:	Rebuild source code dependencies
make clean:	Delete all stale object files

.✪ *Note: Back up the* `/usr/src/linux/.conf` *file to another location and rename it (for example:* `conf_old`*). If you ever have to reconfigure a fresh kernel, copy this file back into the linux directory, rename it to* `.config` *and run* `make menuconfig`*.*

For 2.2.X / 2.4.X kernels,

make bzImage:	Build new kernel.

Important Step

Before proceeding to build new modules, it is advisable to remove existing modules.

- Change directory to `/lib/modules`
- Rename the `<kernel_name>` directory to `<kernel_name>.old`.

  (For example, if working with kernel 2.2.14, then rename the `2.2.14` directory to `2.2.14.old`)

- Change directory back to `/usr/src/linux.`

  make `modules` to build kernel modules,

  make `modules_install` to install modules.

Updating LILO

This section only applies if the kernel was compiled using `bzImage` or `zImage`.

After successful compilation, copy the new image `usr/src/linux/arch/i386/boot/zImage` (or `bzImage`) into the `/boot` directory and rename it to `vmlinuz_2.2.X` (where `X` is the kernel number.) An example appears below:

```
cp /usr/src/linux/arch/i386/boot/bzImage /boot/vmlinuz-2.2.14
```

Note: Before you copy the new kernel into the `/boot` *directory rename the old kernel image*

For example: `mv vmlinuz-2.2.10 vmlinuz-2.2.10.old`

In `/etc` directory, edit the `lilo.conf` file and update the changes. For more info run `man lilo`.

lilo: execute `lilo` to update the changes to `/etc/lilo.conf`

When all this is done, reboot your machine to load the new image.