Embedded Development Tool

LINEO™
Embedix™
SDK

Reference
Manual

LINEO™

Embedix SDK 2.0 Reference Manual

Contents

Contents   **iii**

vi   *Embedix SDK Reference Manual*

Contents   **vii**

About This Guide

This document is addressed to the OEM developer new to the Embedix™ Software Development Kit (SDK) and Embedix Target Wizard.

This preface lists the contents of each chapter, explains the conventions used in this document, provides information about additional resources.

Contents of This Guide

Conventions Used in This Document

The style conventions used in the printed and PDF format of this document do not necessarily apply to other formats. During conversion to HTML, some of these conventions may be lost.

▸ Linux filenames and commands are case-sensitive. In most instances, they are lowercase. When you enter a filename or command, use the same case that appears in the instructions or examples.

▸ All commands or data to be entered on an on-screen data entry line appear in bolded `Courier` font. This may include commands used with options, paths to directories or files, or other simple input, such as filenames.

▸ Other software entities, such as code samples, names of commands, daemons, tools, shells, etc., appear in `Courier` font. Depending on the source of the documentation, variables may appear in all uppercase or in italic font.

▸ When procedures refer to a particular on-screen button, the name of the button appears in uppercase (such as in SAVE), regardless of how it appears on the screen.

▸ When procedures refer to a particular key on a keyboard, only the initial key is capitalized (such as the Tab key), just as it appears on a U.S. standard keyboard. This also applies to key combinations.

▸ Key combinations (such as Ctrl+O) are used in this manner:

1. Press and hold the first key.

2. Press the second key.

3. Then release both keys.

For example, if you are instructed to use Alt+H, you would

1. Press and hold the Alt key.

2. Press the H key.

3. Then release both keys.

▸ All commands or data to be entered in an onscreen data entry line appear in bolded `Courier` font. This may include commands used with options, paths to directories or files, or other simple input, such as filenames.

For example, a URL to be entered in the Web site address field would appear similar to the following:

http://www.lineo.com

‣ Note, Tip, and Warning paragraphs draw your attention to additional information. Ignoring this information may cause a loss of data or time.

Note: Notes contain additional information about the current topic.

Tip: Tips contain suggestions that may save you time or effort.

Warning: Warnings contain critical information that you need to understand before proceeding. Ignoring information in a warning may cause loss of date or time.

Additional Resources

The following resources are available to provide you with additional support in using Embedix SDK. All printed manuals are available in the Embedix SDK box or are with *other* products included in the box.

‣ *Embedix SDK Getting Started*

A printed manual that is also available as softcopy on the CD-ROM in both PDF and HTML formats

‣ *Embedix SDK Target Wizard User Guide*

A printed manual that is also available as softcopy on the CD-ROM in both PDF and HTML formats

‣ *Embedix RealTime Programming Guide*

A printed manual that is also available as softcopy on the CD-ROM in both PDF and HTML formats

‣ Lineo Support Web site

<http://www.lineo.com/support>

1   Embedix Overview

This chapter covers the following topics:

What Is Embedix?

The Lineo™ Embedix™ operating system is a Linux operating system for the embedded systems market. Embedix has a very small memory and disk footprint and can be configured, using tools provided in the Embedix SDK, to meet your specific embedded product needs.

The Embedix operating system includes recent stable versions of the Linux kernel, device drivers, and libraries that provide basic functionality, such as a shell and standard tools.

What Sets Embedix Apart?

The Embedix operating system can provide all the features of a full-featured Linux systems and uses only a fraction of the footprint such as system would require.

Embedix Linux is much more than a desktop distribution that has been forced into a small footprint. The Embedix SDK provides

utilities and programs that have been customized and tuned for use in embedded systems. Several techniques have been used:

▸ Multipurpose programs

▸ Alternate program versions with smaller footprints

▸ Library reduction

Although many distributions of Linux for desktops come with several hundred packages, the Embedix SDK provides a number of specially chosen packages that you can select for your custom Embedix system, including shared libraries, memory protection, multiprocessing, multiuser, UNIX security, networking, etc.

Hardware Requirements for Target Device

Supported Processors

▸ Intel 386 processor or higher

▸ Processor architecture of the included Embedix BSP

Minimal Memory and Disk Requirements

▸ 4 MB RAM

▸ 2 MB ROM, Flash, or Disk

Embedix OS Features

About the Kernel

The Embedix operating system uses a recent and stable version, version 2.4.*x*, of the Linux kernel. For more information about this version of the Linux kernel, see <http://www.kernel.org>.

About Shared Libraries

To reduce the memory and disk footprint of the distribution, the Embedix operating system uses shared libraries where possible.

The Embedix Target Wizard product (part of the Embedix SDK) includes a library reduction tool called LIPO as a build option. The tool analyzes the set of binaries included in the Embedix product to determine which symbols can be removed from the shared libraries. It then produces shared libraries containing a set of symbols that are needed for the Embedix binaries to operate.

About Directory Structure

The directory structure of the root filesystem is shown in Table 1-1. For more information on the standard use of these directories, see the following Web site: <http://www.pathname.com/fhs>..

Table 1-1.
Root Filesystem Directory Structure

/	Root directory
bin	Essential command binaries
boot	Static files of the boot loader and kernel image
dev	Special device files
etc	Host-specific system configuration
home	User home directories
lib	Essential shared libraries and kernel modules

Chapter 1: Embedix Overview   7

Table 1-1.
Root Filesystem Directory Structure *(continued)*

mnt	Mount point of temporary partitions
opt	Add-on application software packages
root	Home directory for the root user
sbin	Essential system binaries
tmp	Temporary files
usr	Secondary hierarchy
var	Variable data
proc	Dynamic information reported by the Linux kernel

Embedix SDK Packages

The Embedix SDK contains packages that have been selected to meet requirements for the majority of embedded systems. For information specific to Embedix on each of these components, see Chapter 6, "Packages," beginning on page 33.

The packages deliver essential files only. Development files and documentation will not be installed to the target.

Note: User manuals and reference documentation for all included packages are included on the Embedix SDK CD-ROM under the `/documents` directory. The reference documentation has been compiled from open source documentation and provided in the SDK for your convenience.

In many cases, documentation for a package is available at the system prompt on the host operating environment.

To access man pages from the Linux prompt on your host system, enter

```
man command_name
```

For example, if you want to learn more about the cron daemon, from the Linux prompt on your host machine, enter

```
man cron.
```

However, if you want to find names of man pages where a keyword—such as a command name, topic, or functionality—is mentioned, you can enter

```
apropos keyword
```

Documentation for programs from the GNU project are often provided by the Info program. To access the info page for a particular command, enter

```
info command_name
```

Licenses Provided

Sample texts of the following licenses are also available in this manual. See Appendix: "License Samples" on page 125

▸ EULA: The end-user license agreement for Embedix is available on the CD-ROM at /LICENSE. This license covers the complete collection of software in this product, and certain other individual components.

▸ GPL: The GNU General Public License covers the Linux kernel and many other software elements in Embedix.

▸ Other licenses may be required by individual packages.

2 Linux Initialization for Intel Platforms

Lineo™ software uses the LILO boot loader to load the kernel into memory on your target *x*86 machine and start the Embedix™ product. The Embedix SDK includes support for LILO.

This chapter, which explains the initialization process, contains the following sections:

▸ "Boot Loader" on page 11

▸ "Initialization Overview" on page 12

Boot Loader

Although LILO can load the kernel from a variety of media, including floppy disks, hard disks, and some flash devices (configured to operate as hard disks), LILO typically loads from disk.

LILO can be configured to pass startup parameters to the Linux kernel during the bootstrap process. Depending on how you configure LILO, you can have the target device wait for input (to select from one of several bootstrap configurations), customize the startup screens for the embedded device, and control other aspects of the initialization and runtime configuration that LILO boots.

LILO can also be configured to load a special filesystem, referred to as an initial RAM disk, or initrd, that can augment the bootstrap sequence in certain ways.

Initialization Overview

When the operating system starts, the kernel runs `/sbin/init`. This can be either a script or a program, but typically it is a program that processes the statements in `/etc/inittab`.

> **Tip:** You can use any `init` program or script that fits your needs. For example, if you have an embedded device for a single custom application, you can replace `/sbin/init` with your application, and the Linux kernel will start your application on boot instead of the original `init`.

The default `inittab` file tells init to run the commands in the file `/etc/rc.d/rcS`. In addition, `init` spawns programs to present login prompts on two virtual terminals.

The operations performed by the `rcS` script are detailed in the next section.

Init System

Embedix uses a modified and much reduced System V init system to start, stop, and control the daemons that run at boot time.

The `init` program used by Embedix is provided by the multifunction program BusyBox. That is, `/sbin/init` is actually a symbolic link to `/bin/busybox`.

Although a typical System V init system establishes multiple run levels (with possibly different sets of daemons started in each run level), but only one run level is available in BusyBox `init`.

The typical filesystem configuration is to mount the root device as read-only. This is because the filesystem is checked for errors during initialization.

When the kernel runs `/sbin/init`, the following events occur:

1. `init` runs the system configuration and initialization script `/etc/rc.d/rcS`.

This script can be used to remount the root device as configurable read/write.

2. The `/etc/rc.d/rcS` script then runs the scripts `rc.modules` and `rc.serial` (if they exist).

3. If configured to do so, `rcS` mounts local filesystems at this point, based on the contents of `/etc/fstab`.

   The /etc/fstab file can also be configured for `rcS` to mount the network filesystems when other init scripts have run.

4. Then `rcS` proceeds to run all the System V init scripts located in the `/etc/rc.d/init.d/` directory, one-by-one in the numeric sequence indicated by their filenames.

   By default, each of the scripts in `/etc/rc.d/init.d/` has an `exit 0` statement at the top of the script. This statement must be commented out for the script to function at startup.

   Some of these scripts also require additional configuration modifications.

Tip: If you customize configuration files, you must place a copy of your modified version of this file in the user merge directory so that Target Wizard can include it in future builds of the target image.

5. Finally, `rcS` runs the commands in the script `rc.local`.

   If you have customizations to the startup commands, you can add or modify either the `rcS` or `rc.local` script.

6. When the `init` process receives a signal to reboot or halt, it sends the `HUP` signal to all processes, sleeps two seconds, sends the `KILL` signal, sleeps one second, and then proceeds either to reboot or to halt.

   The init scripts in `/etc/rc.d/init.d/` are not run with the `stop` command. Each daemon is responsible for handling this behavior properly.

Chapter 2: Linux Initialization for Intel Platforms   **13**

Inittab Format (rcS)

The `init` program reads and processes the file `/etc/inittab` when the kernel starts `/sbin/init`. Statements in this file are interpreted and used to start child processes that implement the functionality required in the target device.

> **Note:** For a general overview of the `init` program, see the `init` man page on the host system. For a detailed description of the `inittab` file, also see the `inittab` man page.

For Embedix, the BusyBox version of `init` operates very much the same as a typical System V `init` system, with a few minor exceptions that are noted here.

▸ The BusyBox version of `init` establishes only one run level at initialization. It does not support multiple run levels. If you need run levels, use `sysvinit`.

▸ The meanings of some of the fields in the `inittab` entries are modified because of some of the special requirements of embedded devices.

In addition, BusyBox init can work without an `inittab` file. If no `inittab` is found, BusyBoxinit assumes the following defaults:

```
::sysinit:/etc/init.d/rcS
::askfirst:/bin/sh
::ctrlaltdel:/sbin/reboot
::shutdown:/sbin/swapoff -a
::shutdown:/bin/umount -a -r
```

If BusyBox init detects that `/dev/console` is *not* a serial console, it will also run:

```
tty2::askfirst:/bin/sh
tty3::askfirst:/bin/sh
tty4::askfirst:/bin/sh
```

A typical inittab entry is a single line of fields with the following format:

```
<id>:<runlevels>:<action>:<process>
```

BusyBox init processes these fields in this manner:

▶ **<id>**: Note that the id field has a nontraditional meaning for BusyBox init. The contents of this field are appended to /dev/ and used as-is. This field does not need to be unique; however, if it isn't, results can be unpredictable. If this field is left blank, it is ignored.

Also note that when BusyBox detects that a serial console is in use, all entries containing nonempty id fields will *not* be run. BusyBox init does nothing with utmp.

▶ **<runlevels>**: init ignores this field.

▶ **<action>**: The action field specifies the action to be taken and the circumstances in which the process for this entry should be run. (Unrecognized actions, such as initdefault, cause init to issue an error message and then to proceed.)

The available actions can be classified into two groups:

▷ Actions that are run only once: sysinit, wait, once, ctrlaltdel, and shutdown

▷ Actions that are re-run when the specified process exits: respawn, askfirst

BusyBox init supports a subset of the actions that are typically supported by init, including those listed in Table 2-1

Table 2-1.
Actions Supported by BusyBox Init

sysinit	The process executes during system boot.
wait	The process executes once after the sysinit actions are processed. Init waits for the process to terminate before proceeding.

Chapter 2: Linux Initialization for Intel Platforms   **15**

Table 2-1.
Actions Supported by BusyBox Init *(continued)*

once	The process executes once after the sysinit and wait actions are processed. Init does not wait for the process to terminate before proceeding.	
respawn	The process restarts whenever it terminates.	
askfirst	Init requests input before starting the process. When the process terminates, init again prompts for input before starting the process.	
	Askfirst functions like respawn, but before running the specified process, it displays the message "Please press Enter to activate this console."	
	The askfirst action is useful in conserving memory on a target device, because it spawns the process only on user input, rather than allowing the process to wait permanently for input.	
ctrlaltdel	The process executes when the system detects that someone on the system console has pressed Ctrl+Alt+Del key combination. Typically, this is because someone wants to reboot the systems.	
shutdown	The process specifies the actions to be taken when `init` is told to reboot.	

‣ **\<process\>**: Specifies the process to be executed and its command line.

Sample /etc/inittab File

All actions are run in the reverse order from where they appear in /etc/inittab.

The following sample /etc/inittab file provides useful examples of entries:

```
# Boot-time system configuration/initialization
# script.
# This is run first except when booting in single-user mode.
#
::sysinit:/etc/init.d/rcS

# /bin/sh invocations on selected ttys
#
# Start an "askfirst" shell on the console (whatever that
# may be)
::askfirst:-/bin/sh
# Start an "askfirst" shell on /dev/tty2-4
tty2::askfirst:-/bin/sh
tty3::askfirst:-/bin/sh
tty4::askfirst:-/bin/sh

# /sbin/getty invocations for selected ttys
#
tty4::respawn:/sbin/getty 38400 tty5
tty5::respawn:/sbin/getty 38400 tty6

# Example of how to put a getty on a serial line
# (for a terminal)
#
#::respawn:/sbin/getty -L ttyS0 9600 vt100
#::respawn:/sbin/getty -L ttyS1 9600 vt100
#
# Example how to put a getty on a modem line.
#::respawn:/sbin/getty 57600 ttyS2

# Stuff to do before rebooting
::ctrlaltdel:/sbin/reboot
::shutdown:/bin/umount -a -r
::shutdown:/sbin/swapoff -a
```

Chapter 2: Linux Initialization for Intel Platforms   **17**

CHAPTER 3

Linux Administration Tasks

This chapter discusses common Linux administration tasks:

▸ "Securing Filesystems" on page 19

▸ "Adding and Removing User Accounts" on page 20

▸ "Adding a Custom Application" on page 22

▸ "Rebooting" on page 24

Securing Filesystems

Because Linux uses file cache by default, some risk of data loss is associated with a sudden power loss to the target computer. This loss typically occurs when part or all of a file in memory has not been written to disk at the time that power is interrupted. Four approaches to dealing with this problem are possible:

1. Disallow writing files.

   Mount all filesystems as read-only. This is an acceptable solution when a computer can be set up once, after which the disk state remains constant.

2. Allow writing files, but when you need to write files—and only then—mount the filesystem as read/write.

   This is a useful solution when files are written infrequently. This can greatly reduce exposure to file corruption.

For example, if you had configured Embedix™ to use a second partition that mounted to /etc and you wanted to change a file under that directory, you would use the following procedure:

2a. Mount /etc as read-only as your default.

2b. When you want to write to a file under /etc, remount that filesystem with the options read/write and sync:

```
mount /dev/xxx /etc -o remount,rw,sync
```

2c. Write the files to disk.

2d. Remount the filesystem as read-only:

```
mount /dev/xxx /etc -o remount,ro
```

The disadvantage to this is that the application writing the file needs to know how to remount the partition.

3. Mount filesystems as read/write and accept the consequences of any interruption.

The sync option to the mount command can be used to reduce the risk somewhat. This solution may be acceptable in circumstances where the computer is in a non-critical application or where backups are made on a regular basis.

4. Use a UPS to supply auxiliary power in the event of power interruption.

There is no support in Embedix to detect power loss.

Choose a method to secure your filesystems, and implement your method using commands in the appropriate initialization scripts.

Adding and Removing User Accounts

Tip: You may not need to support adding and removing users when an embedded device is deployed in the field.

To increase the security for user accounts on the system, Embedix uses a shadow password system. Encrypted user passwords are stored in /etc/shadow instead of in /etc/passwd.

Manipulating Files

Experienced Linux administrators might prefer to add or remove user accounts manually by directly manipulating the following files:

```
/etc/passwd
/etc/group
/etc/shadow
```

Using Commands

Using Linux commands makes the management of users and groups easier and less error-prone.

The following sections explain the Linux commands to use to add and delete users and groups.

Adding Users and Groups

To add a user:

1. Enter the following command:

   adduser *<account_name>*

   where *<account_name>* is the account that you want to create.

   A user ID (uid) is selected automatically by adduser, and an entry for the user is made in /etc/passwd and /etc/shadow. Also, a group is created with the same name as the user account. The group is made the user's primary group, and the user's home directory is created.

2. When prompted, enter a password for the new account, then re-enter the password to verify it.

   The adduser program will report whether the password provided is weak. (Weak passwords are those with too few characters or that are easy to guess, such as passwords that are the same as the account name.)

3. You can control some of the settings for the new account using command line options with adduser. (To view a list of available options, run adduser with no command line arguments.)

4. You can also configure user accounts by the following methods:

   ▷ If you need to change the user ID or group ID for the account, edit /etc/passwd, /etc/shadow, and /etc/group files to reflect your change.

   ▷ To change the ownership of any files owned by this account to the new (correct) user ID, use chown.

   ▷ To change the group ownership of files owned by this account also, use chgrp.

   ▷ To add a group, use addgroup.

Deleting Users and Groups

To remove a user or group account, use deluser and delgroup, respectively. Be aware that when a user account is deleted, deluser does not automatically remove the home directory for the account.

Adding a Custom Application

You can add a custom application to your Embedix build in two ways:

▸ Using Embedix Target Wizard in Embedix SDK (recommended method)

▸ Using an FTP server to transfer the application to your target system

For information on adding an application using Target Wizard, see the chapter "Adding a Custom Application to Your Project" in the *Embedix Target Wizard User Guide*.

We recommend that you follow these guidelines as you develop your application to run on the Embedix product:

▸ You can use any current distribution of Linux as your development platform. Embedix SDK 2.0 is distribution neutral.

▸ Make sure that your application is compiled against the same versions of the libraries provided by the Lineo Embedix product.

If your application is to work with the libraries already provided with Embedix, the most important consideration in developing your application is what libraries (and library versions) your application will use. Embedix provides a nearly complete set of basic system libraries (including libc, libm, and others).

Replacing Core Libraries via Target Wizard

If you need to replace any of the core libraries that are included with Embedix (for example, because it is missing a symbol your application needs), you can do so by transferring the library to the target device.

However, for essential libraries (such as libc), you can't merely copy the new library into place. Instead, you must mount the Embedix partition onto another filesystem (not as the root filesystem) and copy the file when it is not in use.

For example, if you need to copy essential libraries manually:

1. Run Target Wizard.

2. Install the target to the chosen partition.

3. Then exit Target Wizard.

4. Mount the target partition.

5. Copy the libraries to the target partition.

Replacing Core Libraries via FTP Client

When your application and libraries are ready, you can copy them to the Embedix machine using the ftp client and extract them (if needed) using zcat and tar, as in the following example:

```
zcat filename | tar xf -
```

Chapter 3: Linux Administration Tasks   **23**

Rebooting

To reboot Embedix, use Ctrl+Alt+Del from the console. This causes `init` to send a SIGHUP signal to all running processes. After this, `init` sends a SIGKILL signal to all running processes.

You can also reboot the machine from a script or from your own application. In this case, you should execute a call to the program /sbin/reboot from your script or program. This has the same effect (sending SIGHUP and then SIGKILL to all processes).

If you want to stop a particular daemon or service, you can execute its init script with the stop argument. The init scripts are located in /etc/rc.d/init.d/.

For example, to shut down the thttpd Web server, as root enter the following at the shell prompt:

```
/etc/rc.d/init.d/S90thttpd stop
```

4 Loadable Kernel Modules

The Embedix™ kernel supports loadable kernel modules, which are object files that are specially prepared and linked to be loaded and unloaded at runtime in the Linux kernel. Many device drivers and kernel features can be compiled as loadable modules.

This chapter discusses loadable kernel modules in the following sections:

▸ "Why Use Loadable Modules?" on page 25

▸ "Loading and Unloading Kernel Modules" on page 26

Why Use Loadable Modules?

Using loadable modules saves memory and allows you greater flexibility in configuring and customizing your embedded device, even while in the field.

For example, if your target device has an upgradeable network device, you can configure the network driver for the kernel as a loadable module so that you can install a new module in the field without upgrading the entire kernel.

Also, you may want to use loadable modules in case a bug is found in an existing driver. Again, using a module allows you to upgrade only the driver instead of upgrading the entire kernel.

Loading and Unloading Kernel Modules

The Embedix `modutils` package provides user-space utilities that allow kernel modules to be loaded and removed . Two commands in the `modutils` package are used to manipulate modules:

‣ `insmod` for inserting kernel modules

‣ `rmmod` for removing kernel modules

For example, entering **insmod parport** loads the parport kernel module, which allows the use of parallel ports. Entering **rmmod parport** unloads the currently loaded module and disallows the use of parallel ports.

Most modules loaded into the kernel will appear in the /proc directory hierarchy. The proc filesystem mounted on /proc provides system information through a standard virtual filesystem organization. In addition, you can set parameters for some kernel drivers by writing to their corresponding files in the /proc directory.

Load Order

When you load modules, you issue calls to insmod with the names of the modules to be loaded and include any parameters specified during installation.

When you load modules manually or from an autoload script (as explained in the next section), they are loaded sequentially. Because some modules are dependent on the existence of another module or other preloaded kernel support, it is essential to determine load order.

To ensure that required modules are loaded before the modules that depend on them, you can use any of the following ways to determine load order:

‣ Use modprobe. This is an automatic facility for determining Linux kernel module dependencies and module load ordering.

‣ Examine the module load ordering on a host Linux system.

‣ Examine the output from failed module load attempts.

Compare that with symbols in other modules, which you can retrieve using the nm command.

In addition, many modules must be loaded with particular parameters that alter their behavior. You must determine the appropriate parameters for the modules you are loading.

Tip: Allowable module parameters vary from module to module and are documented in Appendix C of *Special Edition Using Caldera OpenLinux.*

Autoloading Modules

You can also configure Embedix to load kernel modules automatically during system initialization.

To do so, add the appropriate insmod commands to /etc/rc.d/rc.modules. Because this file is not created by default, use chmod 755 to create this script.

This file is a shell script that is run by /etc/rc.d/rcS during the initialization sequence, before any package init scripts are executed. Thus, before other daemons are loaded by the system, you can load appropriate network, filesystem, or other device drivers via commands in this file .

Note: As with other configuration files, if you customize this file, you must place a copy of your modified version of it in the user merge directory so that Target Wizard can include the file in future builds of the target image.

Chapter 4: Loadable Kernel Modules   27

CHAPTER 5 Network Configuration and Management

One of the most important features of Linux is its ability to provide networking support. In fact, it may be one reason why you chose the Embedix™ platform for your custom embedded application.

Many excellent books explain Linux networking. This chapter describes only those aspects of network configuration and management that are unique to Embedix.

▸ "Establishing Network Settings"

    ▹ "Assigning Network Addresses"

    ▹ "Configuring Name Resolution" on page 30

    ▹ "Checking the Network Status" on page 31

▸ "Network Devices" on page 32

Establishing Network Settings

To set up basic networking, you must

▸ Assign the IP address and associated network address characteristics

▸ Configure name-to-address resolution service (if one or more applications on the target device require name resolution)

Assigning Network Addresses

The first step in establishing the network settings for the target device is to associate a network address and related address attributes to the device.

The target device can set its network information based on locally stored configuration parameters. This means that the target device

Chapter 5: Network Configuration and Management   29

uses a fixed IP address that is set via commands executed in the file `/etc/rc.d/init.d/S20network`.

To set up a fixed IP address and other network address parameters, edit the file `/etc/rc.d/init.d/S20network`:

1. Comment out "exit 0" at the top of the script.

2. Change the IP address to reflect the network address information for this machine.

   You need to specify the IP address, netmask in Variable Length Subnet Mask (number of bits) format, the network address, broadcast address, and the IP address of a router or gateway machine.

   A subnetted class-C address might appear similar to the following:

   ```
   IPADDR0=207.179.26.19
   NETMASK0=25
   NETWORK0=207.179.26.0
   BROADCAST0=207.179.26.127
   GATEWAY=207.179.26.126
   ```

Configuring Name Resolution

If one or more applications on the target device require name resolution, edit the file /etc/resolv.conf to configure name-to-address service:

1. Add a domain line.

2. Add a search line.

3. Update the address to the name server.

   An /etc/resolv.conf file could contain settings similar to the following:

   ```
   domain mycompany.com
   search mycompany.com
   nameserver 207.179.26.1
   ```

   In either case, whether you are using DHCP (Dynamic Host Configuration Protocol) or manually setting network parameters, set the hostname for your machine by replacing the

line in `/etc/rc.d/rc.local` where the hostname is set. By default this line reads

```
hostname embedix.lineo.com
```

You should change this to some fully qualified domain name for your machine, such as

```
hostname ebxtest.mycompany.com
```

This script should start automatically upon system boot. However, to start the network immediately, at the Linux prompt, execute

`/etc/rc.d/init.d/S20network start`

To stop networking, execute

`/etc/rc.d/init.d/S20network stop`

Checking the Network Status

The ip command allows you to control almost every aspect of your Linux networking. This section, however, introduces only the most common uses of the ip command.

> **Note:** A more detailed description of command features, arguments, and uses of ip (and other Embedix packages) is contained in the technical reference documentation on the Embedix SDK CD-ROM.
>
> In many cases, documentation for a package is available at the system prompt on the host system. For more information, see "Embedix SDK Packages" on page 8.

```
ip address add 127.0.0.1/8 dev lo brd + scope host
ip link set up dev lo
```

▸ Add the IP address 127.0.0.1 with netmask 255.0.0.0 to network device lo (loopback) with broadcast address 127.255.255.255. This address is valid only inside this host.

Chapter 5: Network Configuration and Management   **31**

> ▸ Change the link status on the lo (loopback) device to 'up'.

```
ip addr add 192.168.1.100/24 dev eth0 brd 192.168.1.255
   scope global
ip link set eth0 up
ip route append default via 192.168.1.1 metric 30001
```

```
ip addr show
```

```
ip route show
```

Network Devices

lo	loopback device
eth*x*	Ethernet device
fr*x*	frame relay device
tr*x*	token ring device

```
ip link set down dev eth0
ip addr del 192.168.1.100 dev eth0
```

6 Packages

The Embedix™ SDK provides a number of packages—all selected to meet essential requirements for the embedded market. Embedix packages not only have very small memory and disk footprints, but they have been tuned for embedded systems.

This chapter is organized as follows:

▸ "Overview" on page 34 provides an introduction to the Embedix operating system.

▸ "Package Groups" on page 36 shows the grouping of packages under these general categories:

  ▷ "Core Linux System Packages"

  ▷ "Common Packages"

  ▷ "Embedix-Specific Packages"

  ▷ "Networking Packages"

▸ "Package Descriptions" on page 39. Packages are organized alphabetically by names.

For more information on particular packages, see the `/documents/EmbedixPackagesTechnicalReferences` directory on the Embedix SDK CD-ROM.

In addition, `man` pages or `info` pages are usually available from the system prompt on your host development machine. For more information about `man` pages or `info` pages, see "Embedix SDK Packages" on page 8.

Overview

Shell

The shell included in Embedix, `ash`, is similar to the original Bourne shell `bash`. Although many Linux systems make use of `bash` to provide Bourne-shell functionality, `ash` is much smaller than `bash` and provides a complete set of shell-script commands that are compatible with `bash`. (However, some of the interactive features provided by `bash`, such as tab filename completion, are not available in `ash`.)

Linux Shared Library Loader

To support shared libraries, the system requires the program `ld.so`, the Linux Shared Library Loader.

It maintains a list of libraries that are shared by programs in your implementation of Embedix and tracks which ones have already been loaded into memory. This helps to ensure that your system makes efficient use of memory and disk space with regard to accessed libraries.

When an installed program starts up, `ld.so` ensures that necessary shared libraries are in memory. When a library is called by a second program, ld.so remembers that it already has a copy loaded into memory and it calls the copy.

When a program is closed, `ld.so` determines whether it can close the shared libraries used by that program or whether any of these libraries are required by any other programs still running.

Kernel Binary

The Linux kernel binary is at the core of every Embedix implementation. It handles scheduling, user programs, and resource management. Embedix includes a recent stable version of the Linux kernel (version 2.4).

Kernel modules can be loaded at startup by the script `/etc/rc.d/rc.modules`. See also "Autoloading Modules" on page 27.

Device Files

Linux device files, like their UNIX counterparts, are device pointers that provide a unified communication interface.

For example, if you wanted to access your first floppy drive, you would use `/dev/fd0`; and if you wanted to read the output of your mouse, use `/dev/mouse`.

Editors

Several editors are included in Embedix, such as `ae`, `elvis-tiny`, and `nano`.

For users who are familiar with `vi`, we provide `elvis-tiny`, a smaller replacement for `vi`. It has similar functionality, but features only essential commands. `Elvis-tiny` does not support either split windows or visual markings.

> **Note:** Because of similar functionality, `vi` documentation can provide information that applies to using `elvis-tiny`. In many cases, documentation for a Linux package is available at the system prompt on the host system. For an explanation of online documentation sources, see "Embedix SDK Packages" on page 8.

The `ae` editor is a basic non-`vi` text editor, similar to DOS edit. When using `ae`, a list of available commands is displayed on screen; you can also edit text using menus.

The `nano` editor is a very small and useful editor, formerly known as TIP (TIP Isn't Pico). It aims to emulate `Pico` as closely as possible while also offering a few enhancements.

Package Groups

Core Linux System Packages

In addition to the Linux kernel, these packages are generally considered essential to core Linux operations:

ash
bash
cgetty
cracklib
device nodes
glibc
gpm
libpwdb
lilo
linux
mgetty
modutils
pam-apps
pcmcia
skellinux
util-linux

Common Packages

The following packages provide additional functionality commonly found on full-featured Linux systems:

ae
bzip2
crontab
diffutils
elvis-tiny
ext2fs
file
fileutils

findutils
flex
freetype
gdbm
grep
hdparm
less
libpam
libstdc++
libz
lsof
MAKEDEV (device files)
micro_inetd
microwin (nano-X)
nano
ncurses
pciutils
pidentd
popt
portmap
procps
readline
rtai
sed
setserial
sh-utils
slang
strace
sysklogd
tar
tcp_wrappers
termcap
terminfo
textutils

Chapter 6: Packages   37

time
vixie-cron
which
wu-ftpd
zoneinfo

Embedix-Specific Packages

The following packages are specfic to Embedix because they provide many useful Linux features in a small-footprint system. BusyBox, in particular, contains tiny versions of many popular utilities that are not provided as separate packages.

busybox
default_passwd
tinylogin

Networking Packages

The following packages support Linux networking.

boa
dhcpcd
ipchains
iproute2
iptables
net-tools
netkitbase
netkit-ftp
netkit-telnet
nfs-server
ppp
rsync
thttpd

Package Descriptions

The following section is a comprehensive alphabetical list of the source packages in this product.

ae

Anthony's Editor. A very small text editor that is simple to use. Its default keybindings are similar to `emacs`.

ash

A very small Bourne-compatible shell. In interactive mode, it supports command-line history, but it is other limited in its support for interactive use.

bash

Bourne Again Shell. A Bourne-compatible shell by the GNU project. `Bash` also includes interactive command-line editing, job control, `csh`-like history features, and brace expansion.

boa

A tiny single-tasking Web server. Boa was designed with speed and security in mind aand supports CGI scripts.

busybox

A suite of tiny Linux utilities in a multi-call binary, now upgraded to version 0.51. It is a core tool set that provides common command utilities typically provided on a UNIX system and provides a nearly complete POSIX environment in a very small package.

With a shell such as `ash` and an editor such as `elvis-tiny` or `ae`, you can have a very small but full system. This makes an excellent environment for a rescue disk or any small or embedded system.

The BusyBox command options have been greatly reduced. The included options behave as they do in the GNU versions of these commands that are typically found on a standard Linux installation.

Chapter 6: Packages   39

The utilities included in BusyBox are listed with their descriptions in Chapter 7, "BusyBox."

Note: For the latest information on BusyBox features and acknowledgments to contributors, see the Web site: <http://busybox.lineo.com>

bzip2

A set of file-compression utilities that uses the Burrows-Wheeler block-sorting text-compression algorithm and Huffman coding. The interfaces of these utilities closely mimics those of `gzip`.

cgetty

Console `getty` (get TTY). A login manager that opens and initializes a terminal and prompts the user for a login name. After a login name has been obtained the "login" program is executed. This version ("Console Getty") works for virtual consoles and locally connected text terminals.

cracklib (libcrack)

A program that checks passwords for security-related characteristics—length, uniqueness, whether they are in a word database, etc. This helps ensure that user passwords are not easy to guess. This package is usually used as part of the `libpam` (Pluggable Authentication Modules) library.

crontab

A sample/skeletal `crontab` database for the `cron` daemon .

default_passwd

A set of the default files used for user authentication: `/etc/passwd`, `/etc/group`, `/etc/shadow`, and `/etc/securetty`.

dhcpcd

Dynamic Host Configuration Protocol (DHCP) client daemon. The package provides a DHCP client daemon and related scripts.

DHCP is a standard way of assigning IP addresses from a central computer to one or more clients. The client then configures or reconfigures that network interface for the current system.

1.0.1 DHCPCD is only the client portion of a DHCP connection; if the server is used, it is separately enabled and configured, .

diffutils

A suite of file comparison utilities, including `diff`, `cmp`, `diff3`, and `sdiff`.

The `diff` utilities are used to compare files and then to generate a record of the "differences" between them. This record can be used by the patch program to bring one file up to date with the other. All these utilities (except `cmp`) work only on text files.

elvis-tiny

A text editor that emulates `vi` and `ex`. On systems that pass the program name as an argument, such as UNIX and Minix, you can also install `elvis` under the names `ex`, `vi`, `view`, and `input`. These additional names would typically be links to `elvis`.

ext2fs

A set of tools for creating and maintaining ext2 filesystems.

file

A program that tests each argument in an attempt to classify the file type. There are three tests; filesystem tests, magic number tests, and language tests.

fileutils

A suite of utilities, such as `cp` and `rm`, that are used to manipulate files.

findutils

A suite of utilities, including `find`, `locate`, and `xargs`, that are used to locate files.

flex

Fast Lexical analyzer generator. The GNU project's tool for generating scanners (programs that recognize lexical patterns in text).

freetype

An C library that provides TrueType font support.

gdbm

GNU database manager. A library that provides simple database support for programs.

The `gdbm` module implements an interface to the GNU GDBM database library and builds some higher level facilities on top of it.

gdbserver

A program that allows remote debugging of user space applications on the target system. Used in conjunction with `gdb` provided in the host system's toolchain. For information about kernel debugging, see "startkgdb" on page 50.

glibc

GNU C Library. A collection of essential C functions, both shared and static, that are used by every Linux distribution. Embedix SDK 2.0 currently provides version 2.2.1.

grep

A utility that searches the named input files (or standard input if no files are named, or if the filename "-" is given) for lines containing a match to the given pattern. By default, grep prints the matching lines.

detects the input format.

hdparm

A program that provides a command line interface to various hard disk ioctls supported by the stock Linux IDE/ST-506 device driver.

ipchains

A set of tools for managing Linux kernel packet-filtering capabilities.

iptables

A new set of Linux 2.4 utilities for managing kernel packet-filtering capabilities.

iproute2

A set of tools that provides enhanced IP routing and network devices configuration. It includes the essential tools ip, tc, and rtmon.

The iproute2 package provides ip, the primary tool for manipulating routers. It is used for establishing IP connections and IP links, specifying IP numbers for devices, and for configuring routing tables in the kernel.

Many Linux users are familiar with the command ifconfig. In Embedix, the ip command is used instead of ifconfig and route.

This package includes scripts that establish a default route to bring up an internet connection and to bring it down again.

less

A display pager similar to `more`, but one that allows backward and forward movement. Also, because `less` does not have to read the entire input file before starting, it starts up faster with large input files than `vi`. `Less` uses `termcap` so it can run on a variety of terminals.

libpam

A system of libraries that handle the authentication tasks of services on the system. Provides a pluggable authentication system. See also `pam-apps`.

libpwdb

A standardized interface for programs needing more than a single authentication scheme (for example, `login` needs to know about standard UNIX, shadow passwords, and other schemes).

libstdc++

A package that contains `/usr/lib/libstdc++.so`, a library needed for C++ programs compiled under egcs.

libz

A general-purpose, data-compression library used by many programs.

lilo

Linux Loader. A versatile and powerful boot loader that is used to boot Linux and other operating systems for the Intel 386 PC architecture.

linux

The operating system kernel and modules. Embedix SDK 2.0 is based on kernel version 2.4.

lsof

A utility that lists files opened by processes currently running on the Linux system and provides information about opened files.

MAKEDEV (device files)

The device nodes in the `/dev` directory. Although its name suggests that it is the MAKEDEV shell script, it is instead the actual device nodes themselves, created by that particular version of the MAKEDEV script.

mgetty

A `getty` (get tty) replacement, designed to be used with Hayes-compatible data and data/fax modems. `Mgetty` provides modem initialization, manual modem answering, UUCP locking and very extensive logging facilities.

micro_inetd

A program that listens on the net for requests and spawns a server to handle the request. However, it handles only one port and one program. This daemon would be used instead of `inetd` if you wanted to use it only temporarily or if only one or two daemons need to be controlled. The `inetd` daemon would be a better choice if several daemons need to be controlled.

microwin (nano-X)

Windowing libraries that provide an ultra-small X-Windows server built on top of the frame buffer. `Nano-x` emulates the X11 API; `microwin` emulates the Windows API.

modutils

A set of tools to load, unload, or query kernel modules (drivers, filesystems, etc.) and that dynamically extend the functionality and capabilities of a running kernel.

nano

Nano's ANOther editor. A very small and useful editor, formerly known as `TIP` (TIP Isn't Pico). It aims to emulate `Pico` as closely as possible while also offering a few enhancements.

ncurses

A C library that provides programs with portable, terminal-independent way to update character screens with reasonable optimization. `Ncurses` has support for a wide variety of terminal types.

net-tools

A collection of basic tools used in setting up networking on a Linux system. It includes `ifconfig`, `route`, `netstat`, `rarp`, and some other minor tools.

netkitbase

A collection of networking programs that include the `inetd` daemon, `inetdconf.pl`, and `ping`.

netkit-ftp

A simple client for the File Transfer Protocol (FTP) that is used to retrieve files from FTP servers. This package also contains `pftp` (passive ftp), which is similar to FTP except that it can be used to connect through most firewalls.

netkit-telnet

A small `telnet` client and server.

nfs-server

A suite of server daemons that are needed when you want to export directories on your machine to other hosts via the NFS protocol. It also contains the `showmount` program that can be used to query an NFS server for the list of exported filesystems.

pam-apps

A set of programs that provides user authentication and tools to verify that passwords meet minimum security requirements.

pciutils

A set of utilities for the inspection and setting of devices connected to the PCI bus.

pcmcia

A module that provides core card services, required for all drivers for specific pcmcia cards. It is loaded before loading any socket device drivers.

pidentd

A server that implements the TCP/IP IDENT user identification protocol as specified in the RFC 1413 document. If you make a connection to a service on another machine, that server can retrieve your username by contacting the `identd` daemon running on your machine.

popt

A C library for parsing command line parameters. `Popt` was heavily influenced by the `getopt()` and `getopt_long()` functions, but it improves on them by allowing more powerful argument expansion.

Popt can parse arbitrary `argv[]` style arrays and automatically set variables based on command-line arguments. `Popt` allows command-line arguments to be aliased via configuration files and includes utility functions for parsing arbitrary strings into `argv[]` arrays using shell-like rules.

portmap

A program that manages RPC connections. It converts RPC program numbers into DARPA-protocol port numbers and must be running in order to make RPC calls. When an RPC server is started, it will tell

Chapter 6: Packages   47

portmap what port number it is listening to and what RPC program numbers it is prepared to serve. When a client wishes to make an RPC call to a given program number, it will first contact portmap on the server machine to determine the port number where RPC packets should be sent.

ppp

A set of programs that provide PPP (Point-to-Point Protocol) support for Embedix. PPP is a standard way of establishing serial communications (serial port or modem) between machines.

‣ 1.0.1 PPP is the data framing and handling subsystem of the entire package.

‣ PPPD is responsible for authentication, and handles dial-in access to the Embedix system.

‣ 1.0.2 CHAT is a small communication program that can be used to send short messages to other computer systems.

‣ 1.0.3 PPPSTATS is a utility program that provides PPP connection statistics and other miscellaneous data.

procps

A suite of process monitoring and controlling utilities, such as ps, free, skill, snice, tload, top, uptime, vmstat and w.

readline

A library to which a program can link to provide newline functionality. It can read a line from the terminal and return it, allowing the user to edit the line with the standard emacs editing keys. It allows the programmer to give the user an easier-to-use and more intuitive interface.

rsync

A program that is similar to rcp but which has many more options and uses the rsync remote-update protocol to speed up file transfers when the destination file already exists.

The `rsync` remote-update protocol allows `rsync` to transfer only the differences between two sets of files across the network link because it uses an efficient checksum-search algorithm.

rtai

A real-time application interface developed by the Dipartimento di Ingegneria Aerospaziale del Politecnico di Milano (DIAPM). It allows you to use Linux kernel 2.2.*xx* for many hard real-time applications. This software includes all necessary modules and many examples that can be used to verify proper installation.

For detailed documentation of `rtai`, see the *Embedix RealTime Programming Guide*.

sed

A stream editor that is used for scripted data transformations such as transforming a data stream within a pipeline. It copies the named files (standard input default) to the standard output, edited according to a script of commands.

setserial

A program designed to set and/or report the configuration information associated with a serial port. This information includes which I/O port and IRQ a particular serial port is using, whether the break key should be interpreted as the Secure Attention Key, and so on.

sh-utils

A collection of GNU shell utilities that contain many of the basic commands used for shell programming. Nearly all shell scripts use at least one of these programs.

Shell scripts from other Embedix components may require that you include some of these utilities. Some target projects include shell scripts, (possibly for setup or maintenance) and can require some or all of these utilities.

Chapter 6: Packages   49

skellinux

A skeletal directory structure of a Linux system. See "About Directory Structure" on page 7.

slang

A C library that provides access to a text-console device and allows a program to display character-based graphics on it. It is used by editors, such as `elvis-tiny` and `ae`.

startkgdb

A program that puts the Linux kernel into debug mode. `Kgdb` (often referred to as `gdbstubs`) is a part of the Linux kernel that can be included to allow kernel debugging. `Startkgdb` forces the kernel into debug mode. (Another way to begin kernel debugging is to provide command-line arguments to lilo at boot time.)

> **Warning:** Improperly exiting debug mode when debugging the kernel can result in filesystem damage on the target system.

strace

A debugging utility that traces the system calls that are invoked by a program while the program runs.

syslogd

A general system-logging daemon that is responsible for handling requests for `syslog` services.

Another daemon in this package, `klogd`, runs either standalone or as a client of `syslogd`. `Klogd` listens to kernel log messages, prioritizes them, and routes them either to output files or to `syslogd`.

tar

A general-purpose archive utility that can wrap up whole sets of file trees into a single file. The resulting archive, a tar file (often referred to as a tarball), is often compressed, using `gzip` or some other form of compression. You can recognize tar files by their file extensions: tar.z, tar.gz, tar.Z, tar.bz2, or tar.tgz. Decompression requires `gzip` or `bzip2`.

The `tar` program not only provides the ability to create tar archives, but `tar` makes various other kinds of manipulation possible as well. For example, you can use `tar` on previously created archives to extract files, to store additional files, or to update or list files that were already stored.

`Tar` is most often used in development cycles. It is useful for packaging files together for distribution on other media. (Many embedded products will not require `tar` in their finished form.) Exceptions to this might include devices that provide server or reporting functions where data will be archived and delivered in tarball form or consumer computer devices that require some form of archive program.

tcp_wrappers

A security tool that acts as a wrapper for TCP daemons `tcpd`. The `tcpd` program can be set up to monitor incoming requests for `telnet`, `finger`, `ftp`, `exec`, `rsh`, `rlogin`, `tftp`, `talk`, `comsat`, and other services that have a one-to-one mapping onto executable files.

termcap

A terminal information database that is located under `/etc/termcap`.

terminfo

A small terminal information database located under `/usr/share/terminfo`. It provides information needed by the `slang` library to set up the screen properly for the specified console.

It is smaller than the one included with `ncurses` and contains only a few widely-used terminal types.

textutils

A suite of GNU text-processing utilities. Most of these programs have significant advantages over their UNIX counterparts, such as greater speed, additional options, and fewer arbitrary limits.

A smaller alternative, BusyBox, provides simplified versions of many of these utilities. Consequently, most target systems require only BusyBox. (On the other hand, if the target system requires complete UNIX-style text processing and complete functionality for specific utilities, you should provide textutils for your project.)

thttpd

A very small and very fast Web server that supports most of the functionality of HTTP 1.1, including CGI support, user authentication, content negotiation, throttling, multi-homing, directory indexing, logging, etc. Some additional features, such as server-side includes and document redirection, are available via add-on utilities implemented as CGI scripts. The Web server uses a single-process connection-handling model to avoid extra overhead handling incoming requests.

time

A program that measures many of the processor resources (such as time and memory) that other programs use. The GNU version can format the output in arbitrary ways by using a `printf`-style format string to include various resource measurements. Some systems do not provide much information about program resource use; time will report unavailable information as zero values. It now gets epoch time in seconds.

tinylogin

A Lineo suite of tiny utilities in a multicall binary that provides login functionality. Based on existing Open Source software, `tinylogin` handles user authentication and enables password setting. It works

well with BusyBox (another multi-call binary), and makes an excellent addition to any small or embedded system.

The `tinylogin` commands are

tinylogin	adduser
login	deluser
sulogin	addgroup
passwd	delgroup
getty	su

Note: With each of these commands you can use the `--help` option to get a short description of the available options.

util-linux

A collection of basic, low-level system utilities that are necessary for a functional Linux system. This includes, among other things, configuration tools such as `fdisk` and system programs such as `login`. This package contains miscellaneous programs that do not fit other classifications.

vixie-cron

A `cron` daemon that is used to run specified commands on a periodic basis.

which

A utility that takes a series of program names, and prints out the full pathname of the program the shell needs to call to execute it. It does this by simulating the shell's searching of the $PATH environment variable.

wu_ftpd

Washington University FTP daemon. An FTP daemon that provides added functionality over the standard FTP daemon provided in the `linux-ftpd` package. For example, it can compress files or create tar archives of a directory on the fly; it can provide extensive access control based on the client host address, etc.

zoneinfo

A program that allows you to configure your time zone. `Zoneinfo` uses the `timeconfig` utility to set your locale timezone.

BusyBox

BusyBox combines tiny versions of many common UNIX utilities into a single small executable, providing a fairly complete POSIX environment for any small or embedded system. BusyBox has so many utilities that it requires a separate chapter to describe them all.

This chapter contains the following sections.

▸ "Overview of BusyBox Commands"

▸ "Commands" on page 58

▸ "Compile Time Features" on page 110

Overview of BusyBox Commands

BusyBox provides replacements for most of the utilities typically found in `fileutils`, `shellutils`, `findutils`, `textutils`, `grep`, `gzip`, `tar`, etc. The utilities in BusyBox generally have fewer options than their full-featured GNU counterparts; however, the included options provide the expected functionality and behave very much like the GNU versions.

BusyBox has been written with size optimization and limited resources in mind. To create a working system, just add `/dev`, a shell, and an editor (such as `elvis-tiny` or `ae`) to BusyBox and the Linux kernel.

For a really minimal system, you can even use the BusyBox shell `sh` (not Bourne-compatible, but very small and quite usable), and the BusyBox `vi` editor.

One additional advantage to using BusyBox is that its modularity allows you to include or exclude commands (or even features) at compile time. This means you can configure Busybox to be as small

as possible by selectively turning off any applets you don't need. This makes it easy to customize your embedded system.

Keep in mind that BusyBox commands are documented based on a completely enabled system. When you change features or exclude particular functionality (especially the compile options), the behavior of the BusyBox commands may change.

Warning: Take particular care when disabling any of the compile options.

If a particular command does not function as documented, begin troubleshooting by checking the compile options

What's New or Changed

‣ This upgrade takes BusyBox to 0.51 and includes many new utilities since 0.46.

‣ New utilities:

cmp	dos2unix
dpkg_deb	expr
getopt	ifconfig
pivot-root	rdate
readlink	renice
reset	route
stty	tftp
unix2dos	vi
watchdog	wget
xargs	

Note: The ongoing development of BusyBox is sponsored by Lineo and maintained by Erik Andersen `<andersen@lineo.com>`.

For more information on defined commands as well as acknowledgments to contributors, see `<http://busybox.lineo.com>`.
You can also see man pages for `textutils`(1), `shellutils`(1), etc., from the system prompt of your host development machine.

Synopsis

```
busybox <function> [arguments...]
# or if symlinked
<function> [arguments...]
```

Description

BusyBox combines many common UNIX utilities into a single tiny executable. When you create a link to BusyBox for the function you want to use, BusyBox behaves as if the command itself had been invoked.

For example, if the `ls` command has been compiled into BusyBox, BusyBox behaves as `ls()` if you enter this command:

```
ln -s ./busybox ls
./ls
```

Likewise, you can also invoke BusyBox by issuing the command as an argument on the command line. For example, BusyBox also behaves as `ls()` if you enter this command:

```
./busybox ls
```

Command Options

Most BusyBox commands support the `--help` option to provide a terse runtime description of their behavior.

Commands

Each command or utility that is available in BusyBox, and relevant information regarding each, follows in alphabetical order.

adjtimex

adjtimex [**-q**] [**-o** offset] [**-f** frequency]
    [**-p** timeconstant] [**-t** tick]

Reads and optionally sets system timebase parameters. See adjtimex(2).

Options

-q	Quiet mode - do not print
-o offset	Time offset, microseconds
-f frequency	Frequency adjust, integer kernel units (65536 is 1ppm or .0864 seconds per day). Note that positive values make the system clock run fast
-t tick	Microseconds per tick, usually 10000
-p	Timeconstant

ar

ar -[ovR]{ptx} *archive filenames*

Extract or list files from an ar archive.

Options

-o	Preserve original dates
-p	Extract to stdout
-t	List
-x	Extract
-v	Verbosely list files processed
-R	Recursive action

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 standard output.

Example

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

chgrp

chgrp [OPTION]... GROUP FILE...

Change the group membership of each FILE to GROUP.

Options

-R   Change 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
```

chmod

chmod [**-R**] MODE[,MODE]... FILE...

Change file access permissions for the specified FILE or directory.

Each MODE is defined by combining the letters for who (ugoa) has access to the file, an operator (+ - =) to indicate how permissions will be changed, and a permission (rwxst) for FILE(s) or directories.

Who can be chosen from

u	User who owns the file
g	Users in the file's group
o	Other users not in the file's group
a	All users

Operator can be chosen from

+	Add a permission
-	Remove a permission
=	Assign a permission

Permission can be chosen from

r	Read
w	Write
x	eXecute (or for directories, access allowed)
s	Set user (or group) ID bit

	t	sTicky bit (or for directories, prevents removing files by nonowners)

Options

	-R	Change 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 [OPTION]... OWNER[<.|:>[GROUP]] FILE...

Change the owner and/or group of each FILE to OWNER and/or GROUP.

Options

	-R	Change files and directories recursively

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
```

Chapter 7: BusyBox   **61**

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
```

chvt

chvt N

Change foreground virtual terminal to /dev/ttyN.

clear

clear

Clear the screen.

cmp

cmp FILE1 [FILE2]

Compare files.

cp

cp [OPTION]... SOURCE DEST

Copy SOURCE to DEST or else copy multiple SOURCEs to a directory.

Options

-a	Same as -dpR
-d	Preserve links
-p	Preserve file attributes if possible
-R	Copy directories recursively

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]

Display the current time in the given FORMAT or else set the system date.

Options

-R	Output RFC-822-compliant date string

Chapter 7: BusyBox   **63**

`-d STRING`	Display time described by STRING, not 'now'
`-s STRING`	Set time described by STRING
`-u`	Print or set Coordinated Universal Time (UTC)

Example

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

dc

dc *expression ...*

A tiny RPN calculator that understands the following operations:
`+`, `-`, `/`, `*`, `and`, `or`, `not`, `eor`.

Example

```
$ dc 2 2 +
4
$ dc 8 8 * 2 2 + /
16
$ dc 0 1 and
0
$ dc 0 1 or
1
$ echo 72 9 div 8 mul | dc
64
```

dd

dd [if=name] [of=name] [bs=n] [count=n] [skip=n]
    [seek=n]

Copy a file, converting and formatting according to options.

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 *n* input blocks only
`skip=n`	Skip *n* input blocks
`seek=n`	Skip *n* output blocks

Numbers can be suffixed by w (x2), k (x1024), b (x512), or M(x1024^2).

Example

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

deallocvt

deallocvt N

Deallocate unused virtual terminal /dev/ttyN.

df

df [**-hmk**][FILESYSTEM ...]

Print the filesystem space used and space available.

Options

`-h`	Print sizes in human readable format (for example, 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...]

Strip nondirectory suffix from filename.

Example

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

dmesg

dmesg [**-c**] [**-n** LEVEL] [**-s** SIZE]

Print or control the kernel ring buffer.

Options

-c	Clear the ring buffer's contents after printing.
-n LEVEL	Set console logging level
-s SIZE	Use a buffer of size SIZE.

dos2unix

dos2unix [option] [FILE]

Converts a FILE from DOS format to UNIX format.

Options

-u   Output in UNIX format

-d   Output in DOS format

▸ When no option is given, then input format will be automaticaly detected and converted to the opposite format on output.

▸ When no file is given, then `stdin` is used as input and `stdout` as output.

dpkg_deb

dpkg_deb [**-cexX**] file directory
Perform actions on debian packages (`.debs`)

Options

-c   List contents of filesystem tree (verbose)

-l   List contents of filesystem tree (.list format)

-e   Extract control files to directory

-x   Extract packages filesystem tree to directory

-X   Verbose extract

Example

```
$ dpkg-deb -X ./busybox_0.51_i386.deb /tmp
```

du

du [**-lshmk**] [FILE]...

Summarize disk space used for each `FILE` and/or directory. Disk space is printed in units of 1 KB (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
    (for example, 1K 243M 2G)

Chapter 7: BusyBox   **67**

`-m`	Print sizes in megabytes
`-k`	Print sizes in kilobytes (default)

Example

```
$ ./busybox 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    .
```

dumpkmap

dumpkmap > keymap

Print out a binary keyboard translation table to standard output.

Example

```
$ dumpkmap > keymap
```

dutmp

dutmp [FILE]

Dump utmp file format (pipe-delimited) from FILE or stdin to stdout.

Example

```
$ dutmp /var/run/utmp
8|7||si|||0|0|0|955637625|760097|0
2|0|~|~~|reboot||0|0|0|955637625|782235|0
1|20020|~|~~|runlevel||0|0|0|955637625|800089|0
8|125||l4|||0|0|0|955637629|998367|0
6|245|tty1|1|LOGIN||0|0|0|955637630|998974|0
6|246|tty2|2|LOGIN||0|0|0|955637630|999498|0
7|336|pts/0|vt00anders|anders|:0.0|0|0|0|955637763|0|0
```

echo

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

Print the specified ARGs to stdout.

Options

-n	Suppress trailing newline
-e	Interpret backslash-escaped characters (such as \t=tab, etc.)
-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

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
```

fbset

fbset [options] [mode]

Display and modify frame buffer device settings.

Example

```
$ fbset
mode "1024x768-76"
    # D: 78.653 MHz, H: 59.949 kHz, V: 75.694 Hz
    geometry 1024 768 1024 768 16
    timings 12714 128 32 16 4 128 4
    accel false
    rgba 5/11,6/5,5/0,0/0
endmode
```

fdflush

fdflush DEVICE

Force floppy disk drive to detect disk change.

find

find [PATH...] [EXPRESSION]

Search for files in a directory hierarchy. The default PATH is the current directory; default EXPRESSION is '-print'.

Chapter 7: BusyBox   71

EXPRESSION can consist of

`-follow`	Dereference symbolic links
`-name PATTERN`	Filename (leading directories removed) matches PATTERN
`-print`	`Print (default and assumed)` Print the full filename followed by a newline to stdout
`-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

Example

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

free

free

Display the amount of free and used memory in the system.

Example

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

freeramdisk

freeramdisk DEVICE

Free all memory used by the specified RAM disk.

Example

```
$ freeramdisk /dev/ram2
```

fsck_minix

fsck_minix [**-larvsmf**] /dev/name

Perform a consistency check for Minix filesystems.

Options

-l	List all filenames
-r	Perform interactive repairs
-a	Perform automatic repairs
-v	Verbose
-s	Output super-block information
-m	Activate Minix-like "mode not cleared" warnings
-f	Force filesystem check

getopt

getopt [OPTIONS]...

Parse command options.

Options

-a, --alternative	Allow long options starting with single -
-l, --longoptions=longopts	Long options to be recognized
-n, --name=progname	The name under which errors are reported

`-o, --options=optstring`	Short options to be recognized
`-q, --quiet`	Disable error reporting by `getopt`(3)
`-Q, --quiet-output`	No normal output
`-s, --shell=shell`	Set shell quoting conventions
`-T, --test`	Test for `getopt`(1) version
`-u, --unquote`	Do not quote the output

Example

```
$ cat getopt.test
#!/bin/sh
GETOPT=`getopt -o ab:c:: --long a-long,b-long:,c-long:: \
        -n 'example.busybox' -- "$@"`
if [ $? != 0 ] ; then  exit 1 ; fi
eval set -- "$GETOPT"
while true ; do
  case $1 in
    -a|--a-long) echo "Option a" ; shift ;;
    -b|--b-long) echo "Option b, argument `$2'" ; shift 2 ;;
    -c|--c-long)
      case "$2" in
        "") echo "Option c, no argument"; shift 2 ;;
        *)  echo "Option c, argument `$2'" ; shift 2 ;;
      esac ;;
    --) shift ; break ;;
    *) echo "Internal error!" ; exit 1 ;;
  esac
done
```

grep

grep [**-ihHnqvs**] PATTERN [FILES...]

Search for PATTERN in each FILE or standard input.

Options

-i	Ignore case distinctions
-h	Suppress the prefixing filename on output
-H	Prefix output lines with filename where match was found
-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

This version of grep matches full regular expressions.

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 integrity of compressed file

Example

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

Chapter 7: BusyBox   75

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 to FILE.gz

-d   Decompress

Example

```
$ ls -la /tmp/busybox*
-rw-rw-r--   1 anders anders 1761280 Apr 14 17:47 /tmp /busybox.tar
$ gzip /tmp/busybox.tar
$ ls -la /tmp/busybox*
-rw-rw-r--   1 anders anders 554058 Apr 14 17:49 /tmp /busybox.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 filename. With no FILE, or when FILE is '-', read standard input.

Options

-n N U M   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
```

hostid

hostid

Print a unique 32-bit identifier for the current machine.

hostname

hostname [OPTION] {hostname | **-F** FILE}

Get or set the host name or the DNS domain name. If a host name is given (or a FILE with the '**-F**' parameter), the host name is 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
slag
```

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] ...
```

init

init

Init is the parent of all processes. Embedix uses the BusyBox version of init, which is run only by the kernel and does not support multiple runlevels.

For a complete description of BusyBox init, see "Initialization Overview" on page 12. A sample inittab file is also available on page 17.

insmod

insmod [OPTION]... MODULE [symbol=value]...

Load 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

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`

klogd

klogd `-n`

Kernel logger.

Options

    `-n`   Run as a foreground process

length

length `string`

Print the length of the specified string.

Example

    `$ length "Hello" 5`

ln

ln `[OPTION] TARGET... LINK_NAME|DIRECTORY`

Create a link `LINK_NAME` or `DIRECTORY` to the specified `TARGET`. You can 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 a normal file

Example

```
$ ln -s busybox /tmp/ls
[andersen@debian busybox]$ ls -l /tmp/ls
lrwxrwxrwx    1 root      root 7 Apr 12 18:39 ls -> busybox
```

loadacm

loadacm <mapfile

Load an ACM from standard input. An ACM is a screen-mapping from an application charset to screen font. It is used for internationalization.

Example

```
$ loadacm < /etc/i18n/acmname
```

loadfont

loadfont <font

Load a console font from standard input.

Example

```
$ loadfont < /etc/i18n/fontname
```

loadkmap

loadkmap <keymap

Load a binary keyboard translation table from standard input.

Example

```
$ loadkmap < /etc/i18n/lang-keymap
```

logger

logger [OPTION]... [MESSAGE]

Write MESSAGE to the system log. If MESSAGE is '-', log stdin.

Options

-s	Log to stderr as well as the system log
-f	Log using the specified tag (defaults to username)
-p	Enter the message with the specified priority. This can be numerical or a facility.level pair

Example

```
$ logger "hello"
```

logname

logname

Print the name of the current user.

Example

```
$ logname
root
```

logread

logread

Show the messages from syslogd (using circular buffer).

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 (the time of last modification of file status information)
`-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 tab stop every `NUM` columns
`-t`	With `-l`: show modification time
`-u`	With `-l`: show access time (the time of last access of the file)
`-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 (for example, 1K 243M 2G)
`-k`	Print sizes in kilobytes (default)

Chapter 7: BusyBox   83

lsmod

lsmod

Display a list of all currently loaded kernel modules.

makedevs

makedevs NAME TYPE MAJOR MINOR FIRST LAST [s]

Create a range of block or character special files.

TYPEs include

-b	Make a block (buffered) device
-c or -u	Make a character (unbuffered) device
-p	Make a named pipe. MAJOR and MINOR are ignored for named pipes

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.

Example

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

mkdir

mkdir [OPTION] DIRECTORY...

Create DIRECTORY, if it doesn't already exist.

Options

-m   Set permission mode (as in chmod), not rwxrwxrwx - umask

-p   No error if dir exists; 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
```

Chapter 7: BusyBox   85

mkfifo

mkfifo [OPTIONS] name

Create a named pipe (identical to 'mknod name p').

Options

-m	Create the pipe using the specified mode (default a=rw)

mkfs.minix

mkfs.minix [**-c** | **-l** filename] [**-nXX**] [**-iXX**] /dev/name [blocks]

Make a Minix filesystem.

Options

-c	Check the device for bad blocks
-n [14|30]	Specify the maximum length of filenames
-i INODES	Specify the number of inodes for the filesystem
-l FILENAME	Read the bad blocks list from -l FILENAME
-v	Make a Minix version 2 filesystem

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 (unbuffered) device
`-p`	Make a named pipe. MAJOR and MINOR are ignored for named pipes

Example

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

mkswap

mkswap [**-c**] [**-v0**|**-v1**] device [block-count]

Prepare a disk partition to be used as a swap partition.

Options

`-c`	Check for readability
`-v0`	Make version 0 swap [max 128 MB]
`-v1`	Make version 1 swap [big!] (default for kernels > 2.1.117)
`block-count`	Number of block to use (default is entire partition)

mktemp

mktemp [**-q**] TEMPLATE

Creates a temporary file with its name based on TEMPLATE, which is any name with six `Xs' (such as /tmp/temp.XXXXXX).

Example

```
$ mktemp /tmp/temp.XXXXXX
/tmp/temp.mWiLjM
$ ls -la /tmp/temp.mWiLjM
-rw-------   1 andersen andersen  0 Apr 25 17:10
   /tmp/temp.mWiLjM
```

Chapter 7: BusyBox   87

more

> **more** [FILE ...]

A filter for paging through text one screen at a time.

Example

> ```
> $ dmesg | more
> ```

mount

> **mount** [flags] device directory [**-o** options,
>   more-options]

Mount a filesystem (device) to a specified mount point.

Flags

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

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

async|sync	Writes are asynchronous/synchronous
dev|nodev	Allow/disallow use of special device files
exec|noexec	Allow/disallow use of executable files
loop	Mount a file via loop device
suid|nosuid	Allow/disallow set-user-id-root programs
remount	Remount a currently mounted filesystem, changing its flags
ro|rw	Mount for read-only or read/write

For more filesystem-specific flags, see the mount man page.

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
```

mt

mt [**-f** device] opcode value

Control magnetic tape drive operation.

Available Opcodes

bsf	bsfm	bsr	bss	datacompression
drvbuffer	eof	eom	erase	fsf
fsfm	fsr	fss	load	lock
mkpart	nop	offline	ras1	ras2
ras3	reset	retension	rew	rewoffline
seek	setblk	setdensity	setpart	tell
unload	unlock	weof	wset	

mv

mv SOURCE DEST or:

mv SOURCE... DIRECTORY

Either rename SOURCE to DEST or move SOURCE(s) to DIRECTORY.

Example

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

Chapter 7: BusyBox   **89**

nc

nc [IP] [port]

Mini-netcat that opens a pipe to IP:port.

Example

```
$ nc foobar.somedomain.com 25
220 foobar ESMTP Exim 3.12 #1 Sat, 15 Apr 2000 00:03:02 -0600
help
214-Commands supported:
214-    HELO EHLO MAIL RCPT DATA AUTH
214     NOOP QUIT RSET HELP
quit
221 foobar closing connection
```

nslookup

nslookup [HOST]

Query the nameserver for the IP address of the specified HOST.

Example

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

Name:       debian
Address:    127.0.0.1
```

ping

ping [OPTION]... host

Send ICMP ECHO_REQUEST packets to network hosts.

Options

-c COUNT	Send only COUNT pings
-q	Quiet mode. Display output only at start and when finished

90   *Embedix SDK Reference Manual*

```
-s SIZE        Send SIZE data bytes in packets (default=56)
```

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
```

pivot_root

pivot_root NEW_ROOT PUT_OLD

Move the current root file system to PUT_OLD and make NEW_ROOT the new root file system.

poweroff

poweroff

Shut down the system and request that the kernel turn off power upon halting.

printf

printf format [argument...]

Format and print the given data in a manner similar to the C printf command.

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.

readlink

readlink

Read a symbolic link.

reboot

reboot

Reboot the system.

renice

renice priority pid [pid ...]

Changes priority of running processes.

Allowed priorities range from

  20 (the process runs only when nothing else is running) to

   0 (default priority) to

-20 (almost nothing else ever gets to run).

reset

reset

Resets the screen.

rm

rm [OPTION]... FILE...

Remove (unlink) the FILEs. You can use '--' to indicate that all following arguments are non-options.

Options

-i	Always prompt before removing each destination.
-f	Remove existing destinations; never prompt

Chapter 7: BusyBox   **93**

```
-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.

Options

```
-i              Always prompt before removing each directory.
-f              Remove existing directories; never prompt
-r or -R        Remove the contents of directories recursively
```

Example

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

rmmod

rmmod [OPTION]... [MODULE]...

Unload 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.

rpmunpack

rpmunpack < package.rpm | gunzip | cpio **-idmuv**
Extract an rpm archive.

sed

sed[**-Vhnef**] pattern [files...]

Allowed sed scripts follow this syntax:

'ADDR [!] COMMAND'

where address ADDR can be

NUMBER	Match specified line number
$	Match last line
/REGEXP/	Match specified regexp

(! inverts the meaning of the match)

and COMMAND can be

s/regexp/replacement/[igp]	Attempts to match regexp against the pattern space and, if successful, replaces the matched portion with replacement
aTEXT	Appends TEXT after the pattern space

Options

-V	Print out the version of sed that is being run
-e script	Add the script to the commands to be executed
-f scriptfile	Add the contents of scriptfile to the commands to be executed
-h	Display this help message
-n	Suppress automatic printing of pattern space

Chapter 7: BusyBox   **95**

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.

This version of `sed` matches full regular expressions.

Example

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

setkeycodes

setkeycodes SCANCODE KEYCODE ...

Set entries in the kernel's scancode-to-keycode map, allowing unusual keyboards to generate usable keycodes.

SCANCODE can be either *xx* or e0*xx* (hexadecimal), and KEYCODE is given in decima.l

Example

```
$ setkeycodes e030 127
```

sh

sh [FILE]... or:

sh -c command [args]...

The BusyBox SHell (command interpreter). Use sh just as you would use any other shell. It properly handles pipes, redirects, job control, can be used as the shell for scripts, and has a sufficient set of built-ins to do what is needed. It does not (yet) support Bourne Shell syntax.

If you need things like 'if-then-else', 'while', etc., use ash or bash.

If you just need a very simple and extremely small shell, sh will work very well.

sleep

sleep N

Pause for *n* seconds.

Example

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

sort

sort [-n] [-r] [FILE]...

Sort lines of text in the specified files.

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

For additional settings, see the stty man page.

swapoff

swapoff [OPTION] [DEVICE]

Stop swapping virtual memory pages on the DEVICE.

Options

-a	Stop swapping on all swap devices

swapon

swapon [OPTION] [DEVICE]

Start swapping virtual memory pages on the DEVICE.

Options

-a	Start swapping on all swap devices

sync

sync

Write all buffered filesystem blocks to disk.

syslogd

syslogd [OPTION]...

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

Options

-m NUM	Change the mark timestamp interval; (default=20minutes; 0=off)
-n	Run as a foreground process. Do not fork into the background (when run by init)
-O FILE[:PORT]	Specify an alternate log file; (default=/var/log/messages)
-L	Log locally and via network logging; (default is network-only)

tail

tail [OPTION] [FILE]...

Print last 10 lines of each FILE to standard output. With more than one FILE, precede each with a header containing the filename. 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 (This version of 'tail -f' supports only one file at a time)
`-q`	Never output headers giving filenames
`-s SEC`	Wait SEC seconds between reads with -f
`-v`	Always output headers giving filenames

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 can 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] [FILE(s)] ...

Create, extract, or list files from a tar file. Note that this version of `tar` treats hard links as separate files.

Main operation mode:

`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

Informative output:

v	Verbosely list files processed

Example

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

tee

tee [OPTION]... [FILE]...

Copy standard input to each FILE and also to standard output.

Options

-a	Append to specified FILEs (does not overwrite)

Example

```
$ echo "Hello" | tee /tmp/foo
$ cat /tmp/foo
Hello
```

telnet

telnet HOST [PORT]

Establish interactive communication with another computer over a network using the TELNET protocol.

test

test EXPRESSION or [EXPRESSION]

Check file types and compare values, returning an exit code that is determined by the value of EXPRESSION.

Chapter 7: BusyBox   **101**

Example

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

tftp

tftp command SOURCE DEST

Transfers a file from/to a tftp server using ``octet'' mode.

Commands:

get   Get file from server SOURCE and store to local DEST

put   Put local file SOURCE to server DEST

When naming a server, use the syntax 'server:file'.

touch

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

Update the last-modified date on (or create) the selected FILEs.

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 anders anders   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
```

true

true

Return an exit code of TRUE (0).

Example

```
$ true
$ echo $?
0
```

Chapter 7: BusyBox   **103**

tty

tty

Print the filename of the terminal connected to standard input.

Options

```
-s
```
Print nothing (return only an exit status)

Example

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

umount

umount [flags] FILESYSTEM|DIRECTORY

Unmount filesystems.

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 (that is, 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   Print 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	Print only duplicate lines
-u	Print only unique lines

Example

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

unix2dos

unix2dos [option] [FILE]

Converts FILE from UNIX format to DOS format. Adds a return to every line of a file.

Options

`-u`	Output in UNIX format
`-d`	Output in DOS format

▸ When no option is given, then input format will be automaticaly detected and converted to the oposite format on output.

▸ When no file is given, then `stdin` is used as input and `stdout` as output.

update

update `[options]`

Periodically flush 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 amount of time the system has been running since boot.

Example

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

usleep

usleep N

Pause for N microseconds.

Example

```
$ usleep 1000000
[pauses for one second]
```

uudecode

uudecode `[FILE]...`

Decode a file that has been encoded using uuencode.

Options

```
-o FILE        Direct output to FILE
```

Example

```
$ uudecode -o busybox busybox.uu
$ ls -l busybox
-rwxr-xr-x          1    ams    ams    245264  Jun  7 21:35 busybox
```

uuencode

uuencode `[OPTION] [INFILE] REMOTEFILE`

Encode a file using uucp system.

Options

```
-m             Use base64 encoding as of RFC1521
```

Example

```
$ uuencode busybox busybox
begin 755 busybox
<encoded file snipped>
$ uudecode busybox busybox > busybox.uu
$
```

vi

vi `[OPTION] [FILE]...`

Edit FILE.

Options

```
-R       Read-only. Do not write to the file
```

watchdog

watchdog DEV

Periodically write to watchdog device DEV.

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
```

wget

`**wget** [**-c**] [**-O** file] url`

Retrieve files via HTTP or FTP

Options

-c	Continue retrieval of aborted transfers
-q	Quiet mode. Do not print
-O	Save to filename ('-' for stdout)

which

`**which** [COMMAND ...]`

Locate a COMMAND.

Example

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

whoami

whoami

Print the username associated with the current effective user ID.

Example

```
$ whoami
andersen
```

xargs

xargs [COMMAND] [ARGS...]

Execute COMMAND on every item given by standard input.

Example

```
$ ls | xargs gzip
$ find . -name '*.c' -print | xargs rm
```

yes

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

Repeatedly output a line with all specified STRINGs, or 'y'.

zcat

zcat FILE

Uncompress to stdout.

See "gunzip" on page 75.

Compile Time Features

You can control BusyBox features at compile time:

‣   Include or exclude individual commands within BusyBox

‣   Turn off particularly functionality in the individual commands.

This allows you to achieve an even smaller footprint for your project.

Note: The following options are available within the Target Wizard tree view in the following location: Embedix > System > Utilities > busybox > Enable/Disable compile options?

extra_quiet

```
PROMPT=Enable BB_FEATURE_EXTRA_QUIET?
```

This feature ensures that nothing is printed to the console on boot.

fbset_fancy

```
PROMPT=Enable BB_FEATURE_FBSET_FANCY?
```

This feature turns on extra `fbset` options.

fbset_readmode

```
PROMPT=Enable BB_FEATURE_FBSET_READMODE?
```

This feature turns on `fbset readmode` support.

full_regular_expressions

```
PROMPT=Enable
BB_FEATURE_FULL_REGULAR_EXPRESSIONS?
```

This feature enables full regular expressions. This adds about 4 KB. When this is off, programs—such as `grep` and `sed`—that would normally use regular expressions will use normal strings.

insmod_version_checking

```
PROMPT=Enable
BB_FEATURE_INSMOD_VERSION_CHECKING?
```

This feature enables support of module version checking.

klogd

```
PROMPT=Enable BB_FEATURE_KLOGD?
```

This feature enables `syslogd` to also provide `klogd` support.

ls_recursive

PROMPT=Enable BB_FEATURE_LS_RECURSIVE?

This feature enables `ls` to recursively descend into subdirectories.

mount_force

PROMPT=Enable BB_FEATURE_MOUNT_FORCE?

This feature enables forced filesystem unmounting (for example, in the case of an unreachable NFS system).

simple_ping

PROMPT=Enable BB_FEATURE_SIMPLE_PING?

This feature simplifies the `ping` installation so that it is very small but featureless.

sort_reverse

PROMPT=Enable BB_FEATURE_SORT_REVERSE?

This feature enables reverse sorting.

tar_exclude

PROMPT=Enable BB_FEATURE_TAR_EXCLUDE?

This feature enables support of "`--exclude`" for excluding files.

trivial_help

PROMPT=Enable BB_FEATURE_TRIVIAL_HELP?

This feature compiles out everything but the most trivial '`--help`' usage information (that is, it reduces binary size).

8   Licensing for Embedix Packages

Types of Open Source Licences

The most common Open Source Licenses are listed following:

▸ **GPL:** GNU General Public License

    ▷ Must provide source to customers on request

    ▷ Derivative works must be GPL (open source)

▸ **LGPL:** The Library GPL (also known as the "Lesser GPL")

▸ **BSD:** Berkeley Software Distribution

    ▷ Do whatever you want

    ▷ Keep BSD attribution intact

    ▷ Old style BSD requires a copyright notice in documentation

    ▷ New style BSD does not require copyright notice in documentation

▸ **Artistic:**

    ▷ Allow original artist control of work

    ▷ Used by Perl and others

To view samples of these licenses, see Appendix: "License Samples" on page 125

Rules for Open Source Licenses

These rules apply to all Open Source licenses:

‣ Can charge for any license

‣ GPL—Must provide source to customers on request

‣ Linux Kernel is GPL, but proprietary drivers can be included with the kernel if they are modules.

Licenses Required by Included Packages

The following table lists the required licenses for each package in this product. Most fall under the four common licenses explained earlier in this chapter. Any exceptions are noted in the table.

Table 8-1.
Required Licenses by Package

Package	License Required
ae	GPL
ash	BSD
bash	Free Software Foundation, Inc.; modified GPL
boa	GPL ver. 2
busybox	GPL
bzip2	New style BSD
cgetty	Copyright 1999 by Torsten Duwe (A Caldera employee working in Germany). All Rights Reserved. "This file may be copied under the terms of the GNU Public License." Most of the files, however, either state the BSD license or don't mention a copyright or license at all.
cracklib (libcrack)	Alec David Edward Muffett; Artistic. May be freely copied and used.

Table 8-1.
Required Licenses by Package

Package	License Required
crontabs	Appears to be public domain (Caldera did not include a copyright or license statement).
default_passwd	GPL
dhcpcd	GPL
diffutils	GPL
elvis-tiny	Public domain
ext2fs	GPL
file	Copyright (c) Ian F. Darwin 1986, 1987, 1989, 1990, 1991, 1992, 1994, 1995 Redistributions of source code must retain the above copyright
fileutils	GPL
findutils	GPL

Table 8-1.
Required Licenses by Package

Package	License Required
flex	Flex carries the copyright used for BSD software, slightly modified because it originated at the Lawrence Berkeley (not Livermore!) Laboratory, which operates under a contract with the Department of Energy:
	Copyright (c) 1990 The Regents of the University of California. All rights reserved. This code is derived from software contributed to Berkeley by Vern Paxson. The United States Government has rights in this work pursuant to contract no. DE-AC03-76SF00098 between the United States Department of Energy and the University of California.
	Redistribution and use in source and binary forms are permitted provided that: (1) source distributions retain this entire copyright notice and comment, and (2) distributions including binaries display the following acknowledgement: "This product includes software developed by the University of California, Berkeley and its contributors" in the documentation or other materials provided with the distribution and in all advertising materials mentioning features or use of this software. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
	THIS SOFTWARE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.
	This basically says "do whatever you please with this software except remove this notice or take advantage of the University's (or the flex authors') name."
	Note that the "flex.skl" scanner skeleton carries no copyright notice. You are free to do whatever you please with scanners generated using flex; for them, you are not even bound by the above copyright.

Table 8-1.
Required Licenses by Package

Package	License Required
freetype	Copyright 1996-1999 by David Turner, Robert Wilhelm, and Werner Lemberg "This license was inspired by the BSD, Artistic, and IJG (Independent JPEG Group) licenses" ----------------------------- NOTE ----------------------------- ------ Extracted from http://www.freetype.org/patents.html ------- ----------------------------- NOTE ----------------------------- We recently discovered that Apple owns several patents related to TrueType. A simple advanced search on IBM's Intellectual Property Network website (http://www.patents.ibm.com/advquery) shows that Sampo Kaasila, who were the original TrueType architect at Apple, was granted 5 patents for Apple related to digital font technology. Three of them seem to relate directly to the TrueType specification : Patent #1 : US5155805: Method and apparatus for moving control points in displaying digital typeface on raster output devices Filed on May, 8 1989 Patent #2 : US5159668: Method and apparatus for manipulating outlines in improving digital typeface on raster output devices Filed on May, 8 1989 too. Actually, the two patents were filed and granted concurrently. Patent #3 : US5325479: Method and apparatus for moving control points in displaying digital typeface on raster output devices filed on May 28, 1992 which is the continuation of patent #1. The difference with this patent are extremely subtle, and we fail to see what it covers which isn't in patent #1. ... it affects the bytecode interpreter used to hint TrueType outlines. It also affects any other similar engine that render TrueType fonts per se the specification. Note that the TrueType specification used to write FreeType doesn't mention any patent, nor any pending patents. We used the "TrueType Font Format Specification" document, version 1.0, published in 1990 and available from Apple under the reference "ADPA M0825LL/A". None of the successive releases of this paper document, be they in paper or electronic forms, mentioned them either. (And yes, we're speaking of the documents produced by both Apple and Microsoft). ----------------------------- NOTE ----------------------------- ------ Extracted from .../BUILD/freetype-1.3/PATENTS ----------------------------- NOTE ----------------------------- WE HAVE CONTACTED APPLE'S LEGAL DEPARTMENT AND ARE STILL WAITING FOR THEIR ANSWER ON THE SUBJECT. ----------------------------- NOTE ----------------------------- ------ Extracted from .../BUILD/freetype-1.3/announce ----------------------------- NOTE ----------------------------- It's a clean-room implementation that is not derived from the original TrueType engine developed by Apple and Microsoft, though it matches it regarding rendering quality. To our knowledge, it's the only royalty-free complete TrueType engine available.

Chapter 8: Licensing for Embedix Packages **117**

Table 8-1.

Required Licenses by Package

Package	License Required
gdbm	GPL
glibc	LGPL
gpm	GPL (Version 2, June 1991)
grep	LGPL
hdparm	GPL
ipchains	GPL
iptables	GPL
iproute2	From the spec file: Distributable From the Internet: http://rufus.w3.org/linux/RPM/suse/6.3/i386/suse/n1/iproute2-2.2.4-27.i386.html The copyright is listed as GPL.
less	GPL (Version 2, June 1991)
libpam	Redistribution and use in source and binary forms of Linux-PAM, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain any existing copyright notice, and this entire permission notice in its entirety, including the disclaimer of warranties. 2. Redistributions in binary form must reproduce all prior and current copyright notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. The name of any author may not be used to endorse or promote products derived from this software without their specific prior written permission. ALTERNATIVELY, this product may be distributed under the terms of the GNU General Public License, in which case the provisions of the GNU GPL are required INSTEAD OF the above restrictions. (This clause is necessary due to a potential conflict between the GNU GPL and the restrictions contained in a BSD-style copyright.)
libpwdb	GPL

Table 8-1.
Required Licenses by Package

Package	License Required
libstdc++	GPL or LGPL
libz	(C) 1995-1998 Jean-loup Gailly and Mark Adler. Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions: 1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required. 2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software. 3. This notice may not be removed or altered from any source distribution.
lilo	GPL
linux	GPL
lsof	Copyright 1997 Purdue Research Foundation, West Lafayette, Indiana 47907. All rights reserved. Permission is granted to anyone to use this software for any purpose on any computer system, and to alter it and redistribute it freely, subject to the following restrictions: 1. Neither the authors nor Purdue University are responsible for any consequences of the use of this software. 2. The origin of this software must not be misrepresented, either by explicit claim or by omission. Credit to the authors and Purdue University must appear in documentation and sources. 3. Altered versions must be plainly marked as such, and must not be misrepresented as being the original software. 4. This notice may not be removed or altered.
MAKEDEV (device files)	FRS Freely Redistributable Software
mgetty	The voice portions are under GPL. Copyright (c) 1993 Gert Doering. You are permitted to do anything you want with this program—redistribute, use parts of the code in your own programs, ..., but you have to give me credit—Do not remove my name.
micro_inetd	Copyright (C) 1996 by Jef Poskanzer <jef@acme.com>. (Free to use as long as copyright included—binary & source)

Chapter 8: Licensing for Embedix Packages   **119**

Table 8-1.
Required Licenses by Package

Package	License Required
microwin (nano-X)	The Microwindows, Nano-X, and BOGL software are licensed under MPL. Alternatively, the software can be converted to the GNU General Public License, Version 2
modutils	GPL
nano	GPL (Version 2, June 1991)
ncurses	GPL
net-tools	GPL (Version 2, June 1991)
netkit-base	(BSD) Copyright (c) The Regents fo the University of California Copyright (c) Eric P. Allman
netkit-ftp	Copyright FSF, license appears to be GPL
netkit-telnet	BSD
nfs-server	GPL

Table 8-1.
Required Licenses by Package

Package	License Required
pam-apps	Redistribution and use in source and binary forms of SimplePAMApps, with or without modification, are permitted provided that the following conditions are met:
	1. Redistributions of source code must retain any existing copyright notice, and this entire permission notice in its entirety, including the disclaimer of warranties.
	2. Redistributions in binary form must reproduce all prior and current copyright notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
	3. The name of any author may not be used to endorse or promote products derived from this software without their specific prior written permission.
	ALTERNATIVELY, this product may be distributed under the terms of the GNU General Public License, in which case the provisions of the GNU GPL are required INSTEAD OF the above restrictions. (This clause is necessary due to a potential conflict between the GNU GPL and the restrictions contained in a BSD-style copyright.)
pciutils	GPL
pcmcia	Copyright (C) 1998, 1999, 2000 David A. Hinds
	Unless otherwise indicated, this code is distributed under version 1.1 of the Mozilla Public License ("MPL"), included in the LICENSE file.
	Alternatively, these files may be used under the terms of the GNU Public License version 2 (the "GPL"), in which case the provisions of the GPL are applicable instead of the above. If you wish to allow the use of your version of these files only under the terms of the GPL and not to allow others to use your version of these files under the MPL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the GPL. If you do not delete the provisions above, a recipient may use your version of these files under either the MPL or the GPL.
	Some of the client drivers (nmclan_cs.c, 3c589_cs.c, 3c574_cs.c, 3c575_cb.c, ibmtr_cs.c, pcnet_cs.c, smc91c92_cs.c, fmvj18x_cs.c, wavelan_cs.c, wvlan_cs.c, netwave_cs.c, xirc2ps_cs.c, serial_cb.c) contain code written by others, subject to more restrictive (GPL) licensing requirements.

Chapter 8: Licensing for Embedix Packages   **121**

Table 8-1.

Required Licenses by Package

Package	License Required
pidentd	Public domain as stated by Peter Eriksson
popt	Copyright (c) 1998 Red Hat Software
	Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
	The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
	THE SOFTWARE IS PROVIDED "AS IS," WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
	Except as contained in this notice, the name of the X Consortium shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the X Consortium.
portmap	Copyright (c) 1990 The Regents fo the University of California. All rights reserved.
ppp	GPL / Artistic
procps	By default, the BSD-compress algorithm is available for packet compression. The algorithm is apparently covered by US patents 4,814,746 and 4,558,302. It is possible to create a version of ppp without BSD-compression, but this is not directly handled by Target Wizard, and requires source-code edits. All of the code can be freely used and distributed.
readline	No redistribution criteria are mentioned.
rsync	GPL

Table 8-1.
Required Licenses by Package

Package	License Required
rtai	LGPL
sed	GPL
setserial	GPL
sh-utils	GPL
skellinux	presumed GPL
slang	GPL or Perl Artistic License (both are listed)
strace	Distributable if the copyright notice is included. (Custom copyright notice.)
sysklogd	klogd—GPL syslogd—BSD
tar	GPL
tcp_wrappers	Wietse Venema; by permission.
termcap	Public domain
terminfo	GPL
textutils	GPL
thttpd	BSD-ish. It's not exactly BSD but has the same terms (i.e., it must keep copyright and disclaimer intact. Otherwise there are no other restrictions on distribution).
time	GPL
tinylogin	GPL
util-linux	GPL, BSD
vixie-cron	Distributable so long as the author's name is not removed from the source.
which	Distributable

Table 8-1.

Required Licenses by Package

Package	License Required
wu_ftpd	Copyright (c) 1999 WU-FTPD Development Group. All rights reserved. Portions Copyright (c) 1980, 1985, 1988, 1989, 1990, 1991, 1993, 1994 The Regents of the University of California. Portions Copyright (c) 1993, 1994 Washington University of Saint Louis. Portions Copyright (c) 1983, 1995, 1996, 1997 Eric P. Allman. Portions Copyright (c) 1998 Sendmail,I nc. Portions Copyright (c) 1989 Massachusetts Institute of Technology. Portions Copyright (c) 1997 Stan Barber. Portions Copyright (c) 1997 Kent Landfield. Portions Copyright (c) 1991, 1992, 1993, 1994, 1995, 1996, 1997 Free Software Foundation, Inc. Redistributions qualify as "freeware" or "Open Source Software" under the following terms:...
zoneinfo	GPL

APPENDIX

APPENDIX   # License Samples

For your convenience, this appendix includes the texts of the most common Open Source licenses referred to in Chapter 8, "Licensing for Embedix Packages."

‣ **GPL:** See "GNU General Public License (GPL)" on page 125.

‣ **LGPL:** See "GNU Lesser General Public License" on page 132.

‣ **BSD:** See "Berkeley Software Distribution" on page 142.

‣ **Artistic:** See "Artistic License" on page 143.

GNU General Public License (GPL)

GNU GENERAL PUBLIC LICENSE Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those

countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN

IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.> Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items—whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989 Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

GNU Lesser General Public License

GNU LESSER GENERAL PUBLIC LICENSE Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages—typically libraries—of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.> Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990 Ty Coon, President of Vice

That's all there is to it!

Berkeley Software Distribution

Copyright (c) The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Artistic License

The "Artistic License"

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

"Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.

"Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder as specified below.

"Copyright Holder" is whoever is named in the copyright or copyrights for the package.

"You" is you, if you're thinking about copying or distributing this Package.

"Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)

"Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.

2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.

3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:

a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as uunet.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.

b) use the modified Package only within your corporation or organization.

c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

b) accompany the distribution with the machine-readable source of the Package with your modifications.

c) give non-standard executables non-standard names, and clearly document the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this

Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own. You may embed this Package's interpreter within an executable of yours (by linking); this shall be construed as a mere form of aggregation, provided that the complete Standard Version of the interpreter is so embedded.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whoever generated them, and may be sold commercially, and may be aggregated with this Package. If such scripts or library files are aggregated with this Package via the so-called "undump" or "unexec" methods of producing a binary executable image, then distribution of such an image shall neither be construed as a distribution of this Package nor shall it fall under the restrictions of Paragraphs 3 and 4, provided that you do not represent such an executable image as a Standard Version of this Package.

7. C subroutines (or comparably compiled subroutines in other languages) supplied by you and linked into this Package in order to emulate subroutines and variables of the language defined by this Package shall not be considered part of this Package, but are the equivalent of input as in Paragraph 6, provided these subroutines do not change the language in any way that would cause it to fail the regression tests for the language.

8. Aggregation of this Package with a commercial distribution is always permitted provided that the use of this Package is embedded; that is, when no overt attempt is made to make this Package's interfaces visible to the end user of the commercial distribution. Such use shall not be construed as a distribution of this Package.

9. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.

10. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

Index

Symbols

/etc/group 40
/etc/passwd 40
/etc/securetty 40
/etc/shadow 40
/etc/termcap 51
/usr/lib/libstdc++.so 44
/usr/share/terminfo 51

A

addgroup command 22
adduser command 21
ae editor 35
ae package 39
Anthony's editor. *See* ae
Application, adding 22
ar command 58
Archiving 51
ARP tables 31
Artistic License 143
ash package 39
ash shell 34
at package 39
Autoloading modules 27
Automatic network address configuration
    30

B

bash package 39
bash shell 34
Berkeley Software Distribution license 142
/bin/busybox 12
boa package 39
Bourne shell functionality 34
Broadcast address 30
BusyBox 55
   commands 55, 58
    --help option 57
    Web site 57
busybox package 39
bzip2 40
bzip2 package 40

C

cat command 59
cgetty package 40
Changing file ownership 22
chgrp command 22, 59
chmod command 27, 60
chown command 22, 61
chroot command 62
chvt command 62
clear command 62
Commands, BusyBox 55, 58
Common Linux packages 36

Index   **151**

S

/sbin/dhcpcd 30
/sbin/init 12
Search, setting 30
sed command 95
sed editor 49
sed package 49
Serial communications 48
Serial console 15
setkeycodes command 97
setserial package 49
sfdisk command 97
sh 49
Shadow password system 21
Shared libraries
   loader 34
   maximizing use of 7
Shell 34
Shell utilities 49
showmount command 46
sh-utils package 49
SIGHUP signal 24
SIGKILL signal 24
Signals
   HUP 13
   KILL 13
   SIGHUP 24
   SIGKILL 24
simple_ping compile time feature 112
skellinux package 50
slang library 50
slang package 50
sleep command 97
sort command 98
sort_reverse compile time feature 112
Special Edition Using Caldera OpenLinux
    27

Starting and stopping the network 31
startkgdb package 50
Startup parameters 11
stop command 13
strace package 50
strace utility 50
swapoff command 98
swapon command 99
Symbol reduction 7
sync command 99
sysklogd package 50
syslogd command 99
System log daemons 50
System V init 12

T

tail command 99
tar command 23, 51, 100
tar package 51
tar_exclude compile time feature 112
Target device
   memory and disk requirements 6
   transferring custom application to 22
Target Wizard 7, 23
tc command 43
tcp_wrappers package 51
Technical reference 31
tee command 101
Telnet client and server 46
telnet command 101
termcap package 51
terminfo database 51
terminfo package 51
test command 101
Text editor. *See* Editors
Text processing utilities 52
textutils 52