Frugalware 2.0rc2-62-gda08f69 (Rigel) Documentation

Contents

1 Introduction

Before you start to read this document, you should know some important things about how to read it.

1.1 Things that you should really read

First there are some part of this document that you should really read, to understand how Frugalware works and how to administer it.

IMPORTANT REFERENCES TO READ:

- This introduction ;)
- How to use pacman-g2.
- How to manage services.

1.2 Running console commands

Throughout this document, there is boxed text which shows you console output. These are important and require quite some attention since most of the time you are expected to run them and get the same output.

```
$ echo foo bar
foo bar
```

This is how a console log look. Let's look at its details so you understand what it means.

The `echo foo bar` part is what you should type and it's the command. The following line `foo bar` is the output of the previous command.

<<<>>>

You may wonder what differentiates the command from the output. You see that in front of the command there is a `$`. This indicates that it's a command line, but there is more meaning in this symbol. This symbol can change depending on the user privileges required to run the command.

HERE IS THE LIST OF THE COMMON PREFIX FOR THE CONSOLE COMMANDS:

- `$` indicates that any user can run the command. Most of the time it means you have to run it with your own user account.
- `user$` indicates that the specified user's privileges are required to run this command. Usually this is necessary for security reasons.

  You can get an interactive shell for this user, replacing *user* with the desired user name, by issuing:

  ```
  $ su - user
  ```

- `#` indicates that the `root` user's privileges are required to run this command. Usually this is required to manage the system configuration.

  You can get an interactive shell for *root* running:

```
$ su -
```

2 About Frugalware

> Seeing this feast of wonderful code spread in front of me as a working system was a much more powerful experience than merely knowing, intellectually, that all the bits were probably out there. It was as though for years I'd been sorting through piles of disconnected car parts - only to be suddenly confronted with those same parts assembled into a gleaming red Ferrari, door open, keys swinging from the lock and engine gently purring with a promise of power...
>
> — Eric S. Raymond

The aim of creating Frugalware was to help you do your work faster and simpler. We hope you will like it. In this introduction, we would like to answer a few questions which were asked in several interview with Miklos, the founder of the project. You can reach the full list of articles that have been posted about Frugalware here.

2.1 Short

Frugalware is a general purpose Linux distribution, designed for intermediate users (who are not afraid of text mode).

2.2 Long

What branches does Frugalware have?

"We have a -current and a -stable branch. The -current branch is updated daily, and we provide security support for our -stable branch till the next release, for approximately 6 months."

What is "The Frugalware Philosophy" about?

"Briefly: simplicity, multimedia, design. We try to make Frugalware as simple as possible while not forgetting to keep it comfortable for the user. We try to ship fresh and stable software, as close to the original source as possible, because in our opinion most software is the best as is, and doesn't need patching."

What is the license of Frugalware?

"The license of Frugalware itself stands for the license of the buildscripts used for building Frugalware. That source is available under the GPL license here. Frugalware's original init scripts were written by Patrick J. Volkerding, creator of the Slackware Linux distribution. We release out additions under the GPL, but Patrick J. Volkerding's code is still under the BSD license. Frugalware also has a few side projects, like our pacman-g2 package manager, the Frugalware installer an so on. They are available under the GPL license, too. For more info about the license of the packages included in Frugalware, refer to the /usr/share/doc/*/COPYING files."

What package manager does Frugalware use?

"We have our own package manager, called pacman-g2. It stands for the second generation of the pacman-g1 package manager, as it was originally based on Judd Vinet's great work. The packages are simple .tar.bz2 files, pacman-g2 is written in C, unlike Slackware's shellscript-based package manager (which may be rather slow sometimes)."

How does Frugalware manage updating obsolete packages?

"We don't have any standalone program for updating packages as pacman-g2 manages this task too. To update your package database, use `pacman-g2 -Sy`, and to update your packages according the just synchronized package database, you use `pacman-g2 -Su`. To install package `foo` with the necessary dependencies directly from one of our ftp servers, you should issue `pacman-g2 -S foo`. For more information, refer to the pacman-g2 man page."

Is there any community support available for Frugalware?

"We have mailing lists, IRC channels and forums that can be used to communicate with developers or with other users and to get help. You can reach the list of mailing lists available here. The IRC channels are on the Freenode network (server: `irc.freenode.net`), the discussion forums are available here."

Is there any commercial support available for Frugalware?

"No, there isn't for now, and currently it isn't planned, either."

For whom is Frugalware recommended to use?

"Frugalware is designed for intermediate users. Installing Frugalware doesn't require any magic, of course, but you should read some documentation if you don't know what a partition, an MBR (Master Boot Record), etc. is."

How to become a developer?

"Get involved! :) Download the FST (Frugalware Source Tree) using the `repoman upd` command, which is available in the pacman-tools package. Then start to play with the FrugalBuild scripts, for a skeleton, refer to the `/docs/skel` directory. Try to improve them, or write a new one for a currently unsupported program. Then open feature requests in the Bug Tracking System and attach your patches. From this point everything will come naturally to you :)"

What do developers do?

"In short, what they want to, if they play a square game. They may maintain packages: building them if a newer version is available and update the FrugalBuild scripts to work correctly against a newer version. They can contribute a new build script for a previously non-existent package. They write documentation, fix bugs, provides support, or anything else in connection with the Frugalware community. If you want to help us, but you don't want to be a developers, you may help in translating Frugalware to your or other language. And, of course, we happily accept donations. :) More info here."

Who develops Frugalware?

"An amazing group of volunteers, who are motived by the users to do so. They also do it as a hobby, and they are always working on having up to date knowledge to make Frugalware even better for you."

Is Frugalware specialized in a certain purpose?

"No, it's a general purpose distribution, for desktops, mobile computers and servers."

Do you plan to release a live cd?

"Well, we have already a live cd, called FwLive. Currently it supports only i686, but an x86_64 version is also under development. You can find it in the standard release directories."

Does Frugalware support languages other than English?

"Yes, it supports all languages supported by the packages. If the init scripts, the setup or the documentation is not available in your language, then it simply means they haven't yet been translated."

What about Asian languages?

"Frugalware roughly supports Asian languages, but don't expect too much - using UTF8 is not the default where it is possible."

What architectures does Frugalware support?

"Currently we support x86 (Pentium Pro or higher), x86_64 (k8, aka. amd64) platforms and arm"

How are compressed the Frugalware packages ?

"FPM packages were originally .tar.gz packages, then a bit later we migrated to libarchive, which allowed bzip2 compression. Life was good, but then lzma was came, and I added support for libarchive, though others were not really interested in a migration, so we stick to .tar.bz2. A few months ago libarchive got support for the xz format (which is the successor of lzma), so we switched to it. pacman-g2 still support .tar.gz and .tar.bz2 as well, and the package extension is .fpm all the time to make it clear that it's a Frugalware package"

3 Quick reference

3.1 Informations

- Package management: pacman-g2 (command line)
- Linux kernel 2.6 (no 2.4 support)
- The latest documentation is here.
- Hardware requirements and list of supported architectures are in the Installation section of the documentation.

3.2 Features

- Stable releases every 6 months
- Security support for stable releases
- Text mode installation
- Optional graphical installation
- Offline installation, netboot install supported
- Prebuilt CD/DVD/USB hybrid, TFTP images are available
- Localization supported whereever it's possible
- About 5000 source packages and (as of March 2011) 6000 binary packages supported.

4 Installation

4.1 Hardware requirements

Given that the number of selected packages to install makes a lot of difference, there is no general answer. Though the followings are recommended for a default install:

- Fearless attitude towards text mode
- Some kind of installation media or set of downloaded packages

4.1.1 i686

- A recent (read: Pentium 2 or higher) 32-bit Intel - or compatible - CPU
- 256MB of RAM
- 8GB of disk space (1GB for a minimal install)

4.1.2 x86_64

- A 64-bit AMD - or compatible, so EM64T is fine - CPU
- 256MB of RAM
- 8GB of disk space (1GB for a minimal install)

4.1.3 arm

- A Marvell Kirkwood platform (e.g. SheevaPlug, Seagate Dockstar, OpenRD, . . .)
- 32MB of RAM
- 1GB of disk space

4.2 Choosing installation flavor

Depending on your needs, there are different installers with different characteristics. You can choose which fits you the best.

4.2.1 Installing from CD

This image contains only a base system, which means the minimal set of packages so that later from the system you can install any other package. It may be handy in case the network installer does not recognize your network card.

Pros: Quick and easy to install, even if you network card does not work out of the box.

Cons: You need to knowledge on how to extend the installed system to the average requirements.

4.2.2 Installing from DVD

This contains all packages from the main groups.

Pros: a full offline installation is possible.

Cons: Large amount of data must be downloaded, presumably some unnecessary packages too.

4.2.3 USB isohybrid image

The ISO images you would use to burn the CD now also double as the USB image as well. They are installed the same way as the old USB images were. All you do is copy them directly to the USB stick's device node.

Pros: No need to burn any CD, you can reuse the media.

Cons: You have to be able to boot from USB.

Warning
Writing the image to a USB stick will destroy all the data on the drive. Be careful when specifying target devices / partitions othervise you can easily loose data.

The following command will install the image to the USB stick on any recent Linux system:

Important
Pay attention to see what /dev/sdX device your USB stick is, for example by having a look at the contents of the `/dev/disk/by-id/` directory!

```
# dd if=frugalware-<version>-<arch>-cd1.iso of=/dev/sdX
```

You might be able to use a similar tool (like this) on Windows systems as well, but it seems only supports partitions not whole disks. If you can find a way to successfully write an USB image under Windows, please share with us.

4.2.4 TFTP image

This is a floppy image, for a very special case:

- you want to do a network installation
- you don't want to / can't use CDs
- you don't want to / can't boot from an USB stick
- you can boot from a network card, but your BIOS does not supports so
- you have a floppy drive

Pros: In some cases this is the only way you can install Frugalware

Cons: You need a bootable network card and a working TFTP server

4.2.5 Fwbootstrap (self-contained chroot)

This is a tarball which has to be downloaded and unpacked. Mostly useful for developers who can compile packages in this build environment on a non-Frugalware host system.

USAGE EXAMPLE:

1. Download the tarball

```
$ wget ftp://ftp5.frugalware.org/packages/frugalware/pub/frugalware/\
frugalware-stable-iso/fwchroot-<version>-<arch>.tar.bz2
```

2. Unpack it

```
$ tar xvjf fwchroot-<version>-<arch>.tar.bz2
```

3. Enter the chroot.

```
$ cd fwchroot-<version>-<arch>
$ ./fwbootstrap
```

4. Use it (build a package or two)
5. Exit from the shell and fwbootstrap will unmount the necessary dirs for you.

You can get a list of installed packages in the chroot with issuing the *pacman-g2 -Q* command.

4.2.6 A manual bootstrap

So you want a complete Frugalware installed into /mnt/foo. First of all, you must have a running Frugalware where you are able to do

```
# pacman-g2 -Sy core base -r /mnt/foo
```

which installs the core and base pkgs into it. But beware:

```
$ pacman-g2 -Qo /etc/profile.d/lang.sh
No package owns /etc/profile.d/lang.sh
$ pacman-g2 -Qo /etc/fstab
No package owns /etc/fstab
```

so you have to copy or forge them by hand.

A script is available to somewhat automate this bootstrap method.

Note
Manual bootstrap is the only way to install the arm port at the moment. Follow the qemu and real device arm-bootstrap howtos if you need more info.

4.3 Obtaining a source media

A Frugalware installation media can be obtained from several sources. You can download it freely via HTTP, FTP or rsync. You can also grab it via bittorrent, see Linuxtracker for example.

The following examples explains how you can get the iso images. You have to replace respectively `$version$`, `$arch$` and `$media$` to get the wanted iso image.

Via FTP:

```
$ wget ftp://ftp3.frugalware.org/mirrors/frugalware/pub/frugalware/\
frugalware-$version$-iso/frugalware-$version$-$arch$-$media$.iso
```

Via HTTP:

```
$ wget http://www5.frugalware.org/linux/frugalware/pub/frugalware/\
frugalware-$version$-iso/frugalware-$version$-$arch$-$media$.iso
```

Via rsync:

```
$ rsync -avP rsync://rsync4.frugalware.org/ftp/pub/linux/distributions/\
frugalware/frugalware-$version$-iso/frugalware-$version$-$arch$-$media$.iso ./
```

More info and the full list of mirrors can be found at our download page.

4.4 Using packages from CD/DVD

You have a skeleton system installed from CD/DVD, and you want to use the packages from the media afterwards. There are two methods.

First is the easiest, but needs quite a lot of space (and caution not to use pacman-g2 -Scc ;)): mount the media and install all the .fpm's found in frugalware-i686 (or frugalware-x86_64) dir to */var/cache/pacman/pkg*.

Second is a bit more challenging, but more usable. Add a new line to */etc/pacman-g2/repos/frugalware* before the other Server lines:

```
Server =file:///media/dvd/frugalware-i686
```

On x86_64, use this one:

```
Server =file:///media/dvd/frugalware-x86_64
```

The media should be mounted on /media/dvd, or change the Server lines appropriately.

Also you can only install packages then from the given media, so you have to insert the first CD if you install a package from the first CD and so on. This is something you should pay attention for.

4.5 The installation process

Important
Do not worry if you misconfigured something! Just press <Cancel> in the next dialog and you will see the menu. Just go back to the given part and you can reconfigure it.

- After downloading and burning the CDs/DVD, insert the first CD/DVD to your CD/DVD drive, and reboot your computer. In the grub menu, you can disable the framebuffer, if a framebuffer with resolution 1024x768 is not suitable for your graphics card or monitor. After that, grub loads the kernel and the initrd image.
- At the first dialog, you should select your language. If your language is not on the list, you should choose a language fits for you. You can change these options after installing too.
- The next dialog is only a greetings. Just push <Enter>. Now it is time to select your keyboard type. Pick your one, then hit <OK>!
- After selecting your keyboard map, setup searches for installation media automatically.
- If you use a netinstall image follow these sub-steps. Otherwise jump to the partitioning point!

  Note
  These steps sets up your network options during the install. When you finished installing Frugalware the installer will ask for network options again. Those options will be the installed system's options.

  a. Now you should select your connection type. The installer uses the netconfig utility. You can also find the documentation for netconfig in this documentation. See the part called: *Networking*.
  b. After setting up the network you can choose a mirror for downloading the packages. The installer will try other mirrors too. This feature is useful when you have got a fast local mirror or something similar.

- The next step is partitioning. Frugalware setup displays a list of your hard disks, you should choose one of them to partition it with a program. You can select the partitioning program in the next dialog, currently fdisk and cfdisk are included. You should create at least one partition with type *Linux*, and it is recommended to create a swap partition (with type *Linux swap*). The swap size should be 500-1000MB. When you have finished partitioning, press <Continue>.
- The following list displays your swap partitions, here you can choose which swap partitions are allowed to be used by Frugalware. Then setup formats your swap partitions. If you have no swap partition just press <Cancel>!
- In the next window, you should select your root partition first, then you can choose if you wish to format it or keep the existing filesystem on it. After selecting the root partition, you can setup other Linux partitions, optionally format them, and set their mount points. Using a separate partition is supported for /boot, /home, /var, but not for /usr (see here for more info).
- After having your Linux partitions mounted, you should do the same with your DOS/Windows ones. Setup will display a list of them, if any exists. You should simply choose a mount point for them here.
- Now it is the time to select if you want to use expert menus or not. If you choose expert menu after selecting the categories you will be able to pick packages one-by-one from the selected categories. So if you select apps and base the installer will give you a list of packages in apps, when you finished picking the packages you will see the packages in base. After picking them the installation begins.

  If you choose the normal menu (it's the default) then you will only see the groups, but not the individual packages. So after picking the groups installation starts.
- The next step is to select package categories. If you will not use KDE or GNOME, you may probably want to disable them. In most cases, it is not a good idea to disable other categories. If you selected the expert menu you will see the package list after this dialog.

Note
If the group list is empty that means you probably misconfigured your network. Please go back and try to fix it. You can also test your connection if you press Alt+F2 and try to ping some servers.

- Setup will install the packages your selected from the first CD. When it is done, you will be prompted to insert the next Frugalware install. If you have only one disc, feel free to abort installing packages, you can install anything else from the net later.

5 Upgrading from Frugalware 1.9 to 2.0

5.1 Preamble

The aim of this howto is to show how you can upgrade a Frugalware-1.9 (Arcturus) system to Frugalware-2.0 (Rigel).

5.2 Deprecated package removal

Some packages will not gracefully be removed because of strict dependencies. If you encounter a message such as this during a system upgrade:

```
:: foobar-subpkg: requires foobar=1.0.0
```

Then you will need to perform this command for each of the removed packages:

```
pacman-g2 -Rd <PACKAGES>
```

<PACKAGES> must be replaced by the names of each of the removed packages. This must be done prior to the system upgrade.

5.3 pacman-g2

The new release comes with an improved `pacman-g2`, you should install it first:

```
pacman-g2 -Sy pacman-g2
```

5.4 Upgrading the system

Now it's time to upgrade the system itself:

```
pacman-g2 -Su
:: Starting local database upgrade...
```

You will be asked to replace some packages automatically. These are normal and you are expected to answer *Y* to these questions (or just hit ENTER).

After this, the list of to-be-upgraded packages is displayed. Just hit enter and wait. Make some tea, it can take a while. :-)

5.5 Updating config files

`pacman-g2` does not touch configuration files in case you customized them. You should run

```
find /etc -name '*.pacnew'
```

and update each configuration file based on the `.pacnew` version. Once you're done with one, you should remove the `.pacnew` file.

5.6 bind update

The caching example configuration we provide in the `bind` package is updated, and `named.ca` is renamed to `named.root`. If you build your real configuration on top of this example, make sure you update your `named.conf`.

5.7 The reboot

Since the kernel is upgraded, too, you have to reboot your machine.

Done!

6 Basic configuration

6.1 Introduction

After the installation of the packages, Frugalware setup will configure your new Frugalware system. If you installed the packages manually, then you'll have to perform those configuration steps manually.

Note
If any problem occurs, there is a debug console on `tty4`, you can see that by pressing `Alt-F4`. You can switch back by hitting `Alt-F1`.

6.2 GRUB

The first step is to install GRUB onto your hard disk. There are four options here: installing to the MBR, the root partition, a floppy or simply skipping. Installing to the MBR is the good choice if you want Frugalware to manage your computer's booting. The root is a good idea if you want to install GRUB into your root partition. In this case, GRUB will not modify your existing boot manager. Floppy is a good idea for example if you don't have any boot manager installed, but you want to leave your MBR unmodified.

6.3 Kernel modules

After the installation of GRUB, the installer will configure your kernel modules. This means that an information dialog appears, but nothing more.

6.4 Accounts and passwords

After module configuration, you should change the root password. This is very important as there is no default password. If you skip this step, anybody will be able to login as root.

After this step, you can create a regular (also known as non-root) user. It's highly recommended to create one, and log in as a regular user. If a command should be run as root, you should use su or sudo under console, and `gksu` or `kdesu` under X.

6.5 Network

After this, setup will configure your network settings. Setup simply runs the netconfig utility, which is described in the Networking section.

6.6 Timezone

If network installation is done, we should configure the system's time. This means two actions. First, you should decide if the hardware (BIOS) clock is set to Coordinated Universal Time (UTC). If yes, select yes here. If the hardware clock is set to the current local time (this is how most PCs are set up), say no here. If you are not sure what is this, you should answer no here.

6.7 Mouse

The next step is to configure your mouse. The configuration will take effect on the console mouse services (gpm) and on the X server. The setting is done by xconfig later.

6.8 Graphical interface

If you have installed an X server (by default xorg), the setup will run xconfig. For more information on xconfig, see the section Graphical interface (X11).

7 Pacman-G2

7.1 Basics

Frugalware comes with Pacman-G2 package manager. Pacman-G2 is a fork of the not-yet-released cvs version of the complete rewrite of pacman-g1 by Aurelien Foret (the old monolithic pacman-g1 is written by Judd Vinet). See the README for details. If you want to do anything with packages, you'll always have to use the `pacman-g2` command. Here are some basic actions with `pacman-g2`:

Actions usually used with remote installation from an FTP server:

```
# pacman-g2 -Sy
```

Updates the package database. Before searching for packages or installing them from an FTP server, you will have to use this command.

```
# pacman-g2 -Su
```

Upgrades all packages that are currently installed but a newer version of the package is available on the FTP server.

```
# pacman-g2 -Syu
```

The combination of the above two, that is the command most users use daily.

```
$ pacman-g2 -Sup
```

Prints the URL of all packages that pacman-g2 should download. This way you can download the packages anywhere and then just copy them to /var/cache/pacman/pkg. This is very useful if you have limited bandwidth at your computer, but you can access high bandwidth elsewhere.

```
# pacman-g2 -S sendmail
```

Installs sendmail with all of its dependencies from the FTP server. If it conflicts with any package, you will be asked if pacman-g2 is allowed to remove them.

```
$ pacman-g2 -Ss perl
```

Searches in the package database (on the FTP server). This example will probably display the perl package and all perl modules. Regular expression based search is also supported.

Of course, you can treat packages as normal files, and you can manually add/remove/etc them. Here are some examples:

```
# pacman-g2 -U zsh-4.2.1-1.fpm
```

Adds (or if it's already installed, upgrades) the zsh package, which is located in the current directory.

```
# pacman-g2 -R qt
```

Removes the qt package.

```
$ pacman-g2 -Qs perl
```

Shows every installed packages whose name contains the string perl.

Generally, if you want to turn off checking for conflicting files, you should use the -f parameter, and if you want to turn off all dependency checking, you should use the -d switch.

```
$ pacman-g2 -h
```

This displays all the switches we discussed above, and a lot more. Once again, these are only the basics. You can also use pacman-g2 -Sh or similar to get help on a particular task.

Note
Full documentation for pacman-g2 can be reached by issuing `man pacman-g2`.

7.2 Apt - pacman-g2 cross reference

For those who are familiar with the apt package management tool, here is a quick cross-reference.

Action	Apt command	Pacman-G2 command
Refresh the package database:	apt-get update	pacman-g2 -Sy
Upgrade currently installed packages:	apt-get upgrade	pacman-g2 -Su
Install a new package:	apt-get install foo	pacman-g2 -S foo
Remove a package:	apt-get remove foo	pacman-g2 -Rc foo
Search in the full package database:	apt-cache search foo	pacman-g2 -Ss foo
Install a package from a file:	dpkg -i foo.deb	pacman-g2 -A foo.fpm
Clean the package cache:	apt-get clean	pacman-g2 -Sc

8 Networking

8.1 Initializing the network card

In most cases, configuring your network card will be done automatically by udev. This means that during every system boot your network card will be detected, and the necessary modules will be loaded. If you want, you can load your network card's module manually by editing the `/etc/sysconfig/modules` file and put the module in the blacklist by editing `/etc/sysconfig/blacklist`. Configuring any interface on your card will be the task of the netconfig utility. Initializing your card ends here.

8.2 The netconfig utility

Configuring your network settings is done by the netconfig utility.

1. First, we have to give a name to your computer. The name must consist of at least two parts, separated by a dot (.).

2. In the next dialog, you should choose how your machine connects to the network. If you have an internal network card and an assigned IP address, gateway, and DNS, use static to enter these values. If your IP address is assigned by a DHCP server (commonly used by cable modem services, not equal to DSL services), select dhcp. In case you've got a DSL connection (eg. ADSL) choose the dsl option! Finally, if you do not have a network card, choose the lo choice. The lo is also the correct choice if you are using a PCMCIA network card.

   When you set up the network, the first question will be the interface you want to set up. It is usually eth0, but it can differ when you set up wireless interfaces for example. If you set up a wireless card netconfig will also ask your ESSID and encryption key.

   a. If you chose static, you must give your IP address, the netmask of your local network, your gateway address (you may leave it blank) and the IP address of your primary name server (you can add more nameservers later by editing the */etc/resolv.conf* file) and then the configuration is finished.
   b. If you chose dhcp, you can optionally give your dhcp hostname, however, netconfig will not ask more questions about your network, since all other data will be provided by the DHCP server.
   c. If you chose dsl, you must give your username, something like *someone@provider.net*. Then you'll have to specify the network interface (usually eth0) through which the ADSL connection script will try to communicate with your ADSL modem. Then enter your password twice.
   d. If you chose lo, you don't have to answer any questions.

3. Finally, netconfig will write all your network configuration files. If you want to edit your settings by hand, the interface information is stored in the */etc/sysconfig/network* directory. There is only one file there called default in most cases. It's because you can set up more than one profile. It's very useful if you have a laptop so that you can set up options for all networks you use.

8.3 Basic firewall configuration

Frugalware comes with a firewall configuration working out of the box. This allows all outgoing connections, and incoming packets for established connections. It does not allow normal incoming packages for any ports. The firewall configuration is at */etc/sysconfig/firewall*.

Note
You will not find this file if you have not installed iptables package as this is an iptables firewall.

Let's see an example: you would like to allow others to ssh into your computer. Edit /etc/sysconfig/firewall, remove the hashmark (#) from the beginning of the line under the # ssh description, and restart the firewall:

```
# service firewall restart
```

The same applies for Apache or any other services.

If you would like to have any advanced firewall settings, configure your firewall as root with iptables then save your config as root with:

```
# iptables-save > /etc/sysconfig/firewall
```

Warning
It will overwrite your existing configuration! It is strongly recommended to make a backup of */etc/sysconfig/firewall* before saving your settings.

9 Graphical interface (X11)

9.1 Configuring your graphics card

If you install X, a /etc/X11/xorg.conf.d directory will be created for you, containing XOrg configuration fragment files. In most cases the default configuration will be enough for you, but you can place your own fragments there if you want to manually fine-tune some of the settings.

A common problem is to use a keyboard layout different to the default of the locale, for example you have a non-English locale, thus the default keyboard layout isn't English, either, but you want to have such one. In that case you need to edit the evdev configuration:

```
# vi /etc/X11/xorg.conf.d/10-evdev.conf
```

and change the `xkb_layout` option there to `us`, for example.

9.2 3D acceleration, binary drivers

If there is built-in 3d acceleration support for your card in X, UDev will detect the necessary drivers and X will enable support for them.

If you have an NVIDIA card, you probably need the manufacturer's binary drivers. Obtaining the NVIDIA binary driver is fairly simple:

```
# pacman-g2 -Sy nvidia
```

9.3 Allow root login in KDM/GDM

By default, no root login is permitted on the GUI, the recommended way of running graphical programs as root is to use gksu or kdesu.

To enable it anyway, the following lines should be edited:

For KDM (*/etc/kde/config/kdm/kdmrc*)

AllowRootLogin=false

modify to

AllowRootLogin=true

For GDM (*/etc/gdm/gdm.conf*)

AllowRoot=false

modify to

AllowRoot=true

10 Sound

10.1 Configuring the sound card

Frugalware uses the Advanced Linux Sound Architecture (ALSA) subsystem for sound cards. For older applications, the Open Sound System (OSS) compatibility modules are loaded, but Frugalware does not contain native OSS support.

Finding and loading the necessary module for your sound card is fairly simple. The process is mostly the same as setting up your network card. During every boot, the hotplug scripts will detect your sound card, but, of course, you can take the automatically loaded module to blacklist, and load it manually by editing */etc/sysconfig/modules*.

10.2 Volume configuration with alsamixer

By default, your sound card can be very loud. You can use `alsamixer` to set the volume of your card. Use the < and > keys to mute a channel, up and down keys to set the volume and left or right keys to switch to another channel. You can quit alsamixer by hitting the Esc key.

From now, during shutdown, Frugalware saves your settings, but you can store or load them any time with the

```
# service alsa save
```

and the

```
# service alsa load
```

commands.

11 Printing

Frugalware uses the Common Unix Printing System (CUPS) for handling printers and to manage printing.

11.1 Before you start

Here comes a few advice depending on what manufacturer made your printer.

11.1.1 Hewlett-Packard

You need *hpijs* at least, but you can also install *hplip* for advanced HP support. Also if you have got some priter&scanner machine it's a good idea to use *hplip*.

11.1.2 Canon

Most likely you need one of the bjfilter packages. The following list tell you which package you should use.

- bjfilter-2.2: Canon Pixus 550i / 850i / 950i (i550 / i850 / i950) and iP90 Driver
- bjfilter-2.4: Canon Pixus 560i / 860i / 960i (i560 / i860 / i960) Driver
- bjfilter-2.5: Canon Pixus iP3100 / iP4100 / iP8600 (and Pixma iP1000 / iP1500) Driver
- bjfilter: Canon Pixus iP2200 / iP4200 / iP6600D / iP7500 / MP500 Driver

Please report us if your printer does not listed or listed, but in the wrong line!

11.1.3 Epson

If you own an Epson Color InkJet Printer you need the *pipslite* package. After installing the package do not forget to restart *cups* and start the *ekp* daemon!

```
sudo service cups restart
sudo service ekpd start
sudo service ekpd add
```

Note
Till now nobody confirmed that this package actually works.

11.1.4 Samsung

The Samsung printer driver for cups is called *splix*. After installing it and restarting *cups* you will find your printer when you add it in *cups*.

11.2 Configuring the printer

1. Open your favorite Internet browser and go to http://localhost:631. This is the Web interface of CUPS.
2. Select Administration from the top menu. If a username is required, type root, and give your root password.
3. You can do almost everything here in connection with printing. In our example, we will add a new local printer.
4. Click Add Printer, type in a name and optionally fill the Location and Description lines, then click on continue.
5. Select Device, in most cases it is Parallel Port #1 for older models and one of the USB ports for newer ones. I you have got a USB printer cups will write the printer name next to the proper port.
6. On the next page, select your vendor and your printer type (the driver/filter).

To set up a remote Windows share with password, give a string like this for location (the share name is the printer's assigned name on the remote system): *smb://user:passwd@Netbios_Name_or_ip_address/Share_name*

Notice that, when you view the printer configuration, the credentials will not be shown but will be used.

11.3 My printer is not listed

If your vendor or printer type isn't listed in the wizard, you have to check the OpenPrinting site whether if is supported under Linux or not. Usually it's enough to install the proper printer driver (see above) or *gutenprint*. After installing do not forget to restart cups:

```
# service cups restart
```

If it's not on the page mentioned above, then try to Google after. If listed but said to be "paperweight", then there is nothing to do. If it is supported and said to be working on the site, then please file a bug report with your printer details. While we fix the bug, you can install the driver (the ppd) by yourself.

On the left side, select Printer Listings. Then select your device's vendor and proper type. On the results page, select download PPD. After download, there will be a file named someting_that_ends_with.ppd.

Save the PPD file in the directory /usr/share/cups/model/. The PPD file doesn't have to be executable, but it should be world-readable and should have the file extension ".ppd".

If you do not want to search ppd, try to install foomatic-filters-ppds package. It has a bunch of ppd files for various printers.

Then restart the CUPS service: `su -c \'service cups restart\'`. The driver installation is now completed, now you can add your printer via the web interface. A good howto can be found at http://www.linux-foundation.org/en/OpenPrinting/-Database/CUPSDocumentation.

11.4 Multiple pages on a single sheet

This is also known as n-up printing. If an application doesn't support it natively, print the document to a file as PostScript and use psnup:

```
$ psnup -2 print.ps > print2page.ps
```

The first option specifies the number of pages stacked on one physical sheet, the second is the filename of the original one-sided document, and the last is the n-up (two-sided) document. You can then print it with

```
$ cupsdoprint -P nameofprinter foo.ps
```

or open it in your favourite PS viewer.

11.5 Troubleshooting

If something goes wrong, check out CUPS log at */var/log/cups*. There is a verbose error log and an access log, too.

12 The hotplug subsystem

12.1 udev

The /dev directory under Frugalware is a ramdisk. Every device node is created automatically during the system boot by the hotplug subsystem, more specifically, by udev. It means there won't be unnecessary device nodes in /dev, but it also means that if you create a device node manually, it will exist only until the next shutdown/reboot.

If you want to force Frugalware to create a device node "manually" during each boot, you must create a device file under `/lib/udev/devices`: it will be copied on each boot automatically.

The udev needs sysfs, so it will only work with the 2.6.x kernel series. Do not try to run udev on Frugalware with kernel series 2.4.x.

12.2 Pen/Thumbdrives

Pendrives (also known as thumbdrives, or USB keys) are well-supported through the hotplug scripts and udev. If you insert a pendrive into the USB slot, udev will create a device node for it in /dev. Most pendrives contain only one partition and their filesystem is vfat. In most cases, the pendrive will behave like a SCSI disc. It means, you can find the pendrive under /dev/sda and its first partition under /dev/sda1. Adding the following line to /etc/fstab:

```
/dev/sda1 /media/pendrive auto defaults,noauto,user 0 0
```

will allow users to mount their pendrive if the device node exists (if the device is inserted into the slot).

If you use KDE, Gnome or XFCE4 they will handle automatic mounting of such devices. You should not edit /etc/fstab as automounting will not work for you. For blackbox, fluxbox, englightenment, e17 and other smaller window manager users there is `ivman` for automounting, but it may not work as well as in KDE, Gnome, XFCE4. See also the automounting part of the documentation.

12.3 Digital cameras

Typically, there are two types of digital cameras. Some of them support both access methods, others use only one of them. First, most of the cameras can be treated as a pendrive (USB Mass Storage device), you can mount them and copy the pictures from them easily.

Other cameras support the Picture Transfer Protocol (PTP). You can grab the pictures from them (and do lots of other actions) with `gphoto2`, if your model is supported. (If it's not available on your system, a simple `su -c \'pacman-g2 -S gphoto2\'` will install it onto your system.)

12.4 Automounting: D-BUS, HAL and Ivman; Gnome and KDE

D-BUS is a simple IPC (inter-process communication) library based on messages. HAL is a hardware abstraction layer which uses D-BUS. Ivman is based on HAL and uses pmount ("policy mount"), which is a wrapper around the standard mount program which permits normal users to mount removable devices without an existing */etc/fstab* entry.

Ivman is a daemon to automount CD-ROMs and DVDs when inserted in a drive, or play audio CDs or video DVDs automatically. It is 100% userspace, so it is a safe replacement for submount.

If you want to change the default settings, all config files are located in /etc/ivman. They are plain XML files, just read them, everything is quite self-explanatory.

Automounting also happens with KDE and Gnome, but their respective VFS implementation does that, not ivman. Ivman is useful for other windowing systems where is no support for such a feature.

13 The init scripts, bootup

13.1 About the kernel

The Linux kernel is in the `kernel` package. We use as few patches as possible to stay close to the vanilla kernel. We also use splashy instead of well known bootsplash. The kernel contains compiled-in support for most IDE controllers, but all low-level SCSI drivers are compiled as a module. If Frugalware's kernel doesn't contain built-in support for your controller, you can compile your own kernel. Don't worry, it's fairly simple.

1. After setup is finished, before hitting ENTER to reboot, switch to `tty2` by pressing `Alt-F2` and press ENTER to get a shell.
2. Change your root directory to /mnt/target:
   ```
   # chroot /mnt/target
   ```
3. The source of your kernel (with additional patches applied) can be found at `/usr/src/linux`. So go to the `/usr/src/linux` directory and enter the configuration menu by typing `make menuconfig`. Inside it, select the driver you don't want to compile as a module anymore, and exit from the menu with saving changes.
4. Compile your kernel with the `make` command. This may take several minutes.
5. Copy your new kernel to `/boot` by typing the following command:
   ```
   # cp /usr/src/linux/arch/$yourarch$/boot/bzImage /boot/vmlinuz
   ```
   On i686 and x86_64, `$yourarch$` has to be replaced by x86.

13.2 Init scripts and services

In Frugalware, init is provided by systemd, its service files are always called something.service and they are located in /lib/systemd/system. They are used to setup the environment and manage system services.

The services are UNIX daemons that provide various functionality. The spectrum of their actions is very large. Synchronizing your system clock, running your webserver, running the virus scanner, all of these are services and they offer much much more.

In the following examples we will explain how to alter the running state of a given service. You will have to replace `$service_name$` with the wanted service name, for example `crond.service`. As you will see the syntax is simple, and you may get more help looking at the `systemctl` manual doing:

```
$ man systemctl
```

Important
Later in this document you will see how to alter the configuration of these services so that they follow your needs. You should better learn how to control them, but don't be afraid, the syntax is really simple, and you will learn it in less then a minute.

13.2.1 Controlling a service execution

Services can be started, restarted and stopped, so that you can control what your system has to offer.

To start a service, simply do:

```
# systemctl start $service_name$
```

To restart a service, simply do:

```
# systemctl restart $service_name$
```

To stop a service, simply do:

```
# systemctl stop $service_name$
```

As you can see, controlling a service execution is pretty simple.

13.2.2 Controlling a service execution on system boot

Controlling the automatic execution of services on system startup is not much more difficult.

To add a service for automatic execution on system startup, simply do:

```
# systemctl enable $service_name$
```

To delete a service from automatic execution on system startup, simply do:

```
# systemctl disable $service_name$
```

To check if the service is enabled, simply do:

```
# systemctl is-enabled $service_name$
```

13.3 System boot, targets

If you don't pass any extra *init=/path/to/init* parameters to it, the kernel will start /sbin/init as the final step of the kernel boot sequence. According to /etc/systemd/system/default.target, init will run:

1. each service file required by `basic.target`
2. each service file required by the default target. This is set to `graphical.target` by default. Here is the list of available targets:

```
halt.target = halt
emergency.target = similar to 'init=/bin/sh'
rescue.target = single user mode
multi-user.target = multiuser mode (text mode)
graphical.target = multiuser mode, X11 with KDM/GDM/XDM (default Frugalware target)
reboot.target = reboot
```

Note

`emergency.target` has the advantage that you can boot the system without a reboot later.

If X11 is configured, prefdm.service will start one of the desktop managers, as configured in */etc/sysconfig/desktop*.

13.4 GRUB gfxmenu

Frugalware comes with a nice graphical grub menu (thanks to SuSE's gfxmenu developers). If you don't like it, you can disable it by commenting out the gfxmenu initialization line in /boot/grub/menu.lst.

So for example:

Before: `gfxmenu (hd0,5)/boot/grub/message`
After: `#gfxmenu (hd0,5)/boot/grub/message`

13.5 Splashy

Frugalware uses splashy to display a nice splash screen and a progress bar instead of text messages during the boot procedure. Splashy is completely user-space, so there is no need for patching the kernel. If you dislike it or want to switch it off for whatever reason add nosplashy for your kernel parameters in */boot/grub/menu.lst*. For example:

```
kernel (hd0,2)/vmlinuz root=/dev/hda5 ro quiet nosplashy
```

14 How to contribute

If you appreciate our work, please consider contributing. Below are examples of ways in which you can help the Frugalware project. If you want to help in a way that's not described here, please tell us of your idea in an email to the Frugalware users' mailing list, or add an entry to the Frugalware forums.

14.1 Donations of money

Donations of money are welcome and will be used to cover costs such as domain name registration, hosting costs (hardware, bandwidth etc). If you want to donate, please use the "Donation" link on the Frugalware home page.

14.2 Translation

Comprehensive, multi-lingual documentation is very important to us because we want Frugalware to be available to as many people as possible. If you have the required linguistic knowledge, you could help translate various pieces of work. These include our own applications, documentation, web site etc.

14.3 Application packaging

In the Bug Tracking System, are requests for packages, from Frugalware's users. The process of making packages is well documented in the http://frugalware.org/docs/stable/index-devel [Frugalware Developer Documentation], and with some GNU/Linux experience, you could contribute in that way. Existing package maintainers are always available to help you, especially if you're new to packaging.

14.4 Developing

Frugalware has several of its own applications, including: * An ncurses installer; * A GUI installer (fwife); * A GUI package management tool (gfpm); * A command-line package manager (pacman-g2); * A GUI runlevel manager (gservice).

Help in further developing and enhancing these applications is welcome.

14.5 Donating hardware

By sending us some wanted hardware (see donations), you can make testing packages easier, or speed up the package creation process within a specific architecture.

14.6 Artwork

We usually update our artwork (background images, grub splash, desktop manager themes, window manager splashes and so on) for each release. If you are skilled in this area, you're welcome to join the artwork team.

14.7 Support

If you have time and knowledge, monitor the forums, read the mailing list posts, hang around on IRC and try to answer peoples' questions.

14.8 Find bugs

If you find bugs, you can help by submitting well-written bug reports, see the Reporting Bugs section for more info.

15 The Frugalware Bug Reporting HOWTO

15.1 Introduction

The aim of this HOWTO is to explain how to choose a task name and what to include in a feature request/bugreport to help Frugalware developers speed up the process of fixing a bug or fulfilling a feature request.

15.2 Where

The URL of our Bug tracking system is:

```
http://bugs.frugalware.org/
```

15.3 General

Use the search function before opening a task, as there may be a task for your bug/feature. In that case just add a comment such as "I can reproduce this, too." or "I would enjoy this feature, too."

There are a few topics which are often requested/reported but we have a good reason not fixing/implementing them. You can see a list of such topics in the wiki.

If you'd like to report an outdated package, first check that it is not already listed on this site. If the package is listed, please do **not** report it as we know there is a new version. We will update it as soon as possible in this case.

Write bugreports in English, please. This is the only language all developers speak.

15.4 Bug reports

Please include the following things, unless you know what you are doing:

1. Description of Problem - never say "does not work", quote the error message
2. Steps to reproduce the problem
3. Actual Results
4. Expected Results
5. How often does this happen?
6. Additional Information

The default arch is i686 and the default version is -current. If these are not true, don't forget to change them!

If you report a -current installer bug, then -current is probably not enough, please specify the snapshot date.

If you found a security bug, then use the [SEC] prefix in the task name.

15.5 Feature Requests

Please don't request more than one package in a feature request. Open a task for every package. (Of course you don't have to open task for dependencies if they are also missing from our packages, but please list them if you can.)

If you request a package, please include:

1. The name of the application (yes, "more games" is not enough!)
2. The URL of the application
3. Optionally a short note about why you think this package would be interesting for others, too

If you have a FrugalBuild for the package already, then upload it as an attachment after opening the task. In this case, please prefix your task name with `[FB]`, because this way it will be reviewed sooner.

Alternatively, you can post your FrugalBuild to the `frugalware-devel` mailing list for review, that can be handy if you want to submit more and more buildscripts - finally to become a developer if possible. Opening a task for your FrugalBuild is still fine if you want us to maintain it after the initial version is accepted.

Please don't link other distribution's buildscripts when you request a package. That information is useless for us in most cases and if you don't include such links, you make our life easier.

15.5.1 Do not request

Please don't request custom kernels. We try to use as few patches as possible. See `man kernel.sh` as a reference on building your own kernel using various patchsets. A tutorial is available as well. Really, building such a kernel usually requires a buildscript of only 5 (five) lines!

15.6 Pacman-g2 problems

If you get a crash from our package manager, then we need a backtrace from gdb. Here are the instructions to get a backtrace:

- Find the command line that triggers the crash. For example: pacman-g2 -Sy
- Get the pacman-g2 git repo and compile it with debug symbols enabled:

```
$ git clone http://frugalware.org/git/pub/other/pacman-g2/pacman-g2
$ cd pacman-g2
$ sh autogen.sh
$ ./configure --enable-debug
$ make
```

- Then run pacman-g2 in gdb and get the trace:

```
$ cd src/pacman-g2
$ sudo libtool gdb ./pacman-g2
> run -Sy
```

- When pacman-g2 crashes, get the trace by typing *bt*. Here is an example:

```
Program received signal SIGSEGV, Segmentation fault.
0x0805035e in pacman_sync (targets=0x0) at sync.c:354
354             *p = 1;
(gdb) bt
#0  0x0805035e in pacman_sync (targets=0x0) at sync.c:354
#1  0x08054594 in main (argc=2, argv=0xbfee1844) at pacman.c:609
```

- Attach the output of *bt* to your bugreport.

15.7 Fixed in git

Your feature request/bugreport may be closed with a "Fixed in git . . . " message. Git is our source control management software (just like CVS). If your task is not considered to be critical, then it will be fixed/implemented only in git, without increasing the package release. This means that it will be automatically included in the next release.

16 Mobile computers

16.1 Battery, buttons, thermal management

Notebook users are usually interested in the state of their battery. Getting the power button and the lid's sensor of its closed state to emit events is also nice. Some notebooks only shut down their continously running fans and operate only if needed if the thermal module is loaded.

Usually these modules are automatically loaded by `udev`. If it does not do so for you, then add the following lines to */etc/sysconfig/modules* to get modules loaded at system startup:

```
battery
ac
button
thermal
```

The next task is to enable the acpid service:

```
# service acpid add
```

Then the easiest way is to reboot, or if you don't want do do so:

```
# modprobe battery
# modprobe ac
# service hald stop
# service dbus stop
# service acpid start
# service dbus start
# service hald start
```

The only remaining task is to start a client: if you're on console, try the *acpi* command, or the relevant applet of your favorite window manager.

16.2 Conserving power

The major consumers of power in a notebook are the LCD (differences in size and brightness level can mean a lot), the CPU, hard drives, wireless transceivers like WiFi, Bluetooth, Infrared and the GPU if you have a powerful one.

You can conserve a fair amount of power if you lessen the brightness level of the LCD screen. Some notebooks can remember two settings of this level, one when the equipment operates from battery and another when powered from AC.

The CPUs have some sort of power saving capabilities, the most basic is "CPU throttling". Common on Intel mobile Celeron CPUs, only ACPI is needed. Klaptop has a setting for it, where you can specify the level.

Letting the HDD spin down gives little extra battery operating time, but frequent spinups (data access) and spindowns wears the disk. This is only useful in situations where there is no frequent need for data on HDD like holding a presentation.

16.3 Hibernation

Hibernating your computer can cause data loss or severe filesystem damage if things go wrong. It's highly advised that first, you should consider if hibernating is worth the effort at all. Try it on a fresh installation first, instead of a production system.

From kernel/suspend.c:

```
* BIG FAT WARNING ********************************************************
*
* If you have unsupported (*) devices using DMA...
*                              ...say goodbye to your data.
*
* If you touch anything on disk between suspend and resume...
*                              ...kiss your data goodbye.
*
* If your disk driver does not support suspend... (IDE does)
*                              ...you'd better find out how to get along
*                                 without your data.
*
* If you change kernel command line between suspend and resume...
*                              ...prepare for nasty fsck or worse.
*
* If you change your hardware while system is suspended...
*                              ...well, it was not good idea.
*
* (*) suspend/resume support is needed to make it safe.
```

You have been warned. If you are still not discouraged, read on!

First, you need to create a swap partition (if you don't have any yet). You have to add an extra resume=/dev/swappart kernel parameter to */boot/grub/menu.lst*. For example, on my machine the old line was:

```
kernel (hd0,2)/boot/vmlinuz ro root=/dev/hda3 quiet
```

The new line:

```
kernel (hd0,2)/boot/vmlinuz ro root=/dev/hda3 quiet resume=/dev/hda2
```

After the above are done, you must reboot. The hibernation can be started with:

```
echo shutdown > /sys/power/disk;echo disk > /sys/power/state
```

and next time you boot your kernel it should resume. For more info, look at */usr/src/linux/Documentation/power/swsusp.txt*. It requires the kernel documentation, which can be installed issuing the `pacman-g2 -S kernel-docs` command as root.

17 Packages

The following sections describe the configuration of some packages.

17.1 acoc

In order to use acoc you should start it with

```
$ acoc <command>
```

for example, or you can create an alias like this:

```
alias pacman='acoc pacman'
```

17.2 amavisd-new

For the first initial setup you may want to use our `amavisconf` utility.

From amavisd-new-2.5.2-1 we no longer use a random uid/gid, but dedicated ones. Because of this amavis service will not start if you have it installed before, so you have to correct this by issuing these commands:

```
groupmod -g 40 amavis
usermod -u 40 -g 40 amavis
chown -R amavis:amavis /var/lib/amavis
chown -R amavis:amavis /var/lock/amavis
```

You should chown any other amavis-owned stuff you may have lying around, these are only the default ones.

17.3 android-sdk

Setting up Android SDK :

```
# repoman upd
# repoman merge android-sdk
# pacman-g2 -A android-sdk-r11-1-i686.fpm
```

You should open a new shell to have android-sdk/tools/ in the path. After that, just type "adb" (not "./adb") as mentionned in following links.

If you want to use your Android phone as a proxy, see these pages :

- with Proxoid : http://code.google.com/p/proxoid/wiki/installationLinux
- Proxoid for french users/HTC G1 : http://blog.archambeau.info/?p=9
- with Tetherbot : http://graha.ms/androidproxy/

17.4 apache

17.4.1 How to configure Apache

1. These steps require root privileges, so use su - to get a root shell.
2. The Apache server isn't started by default. You can change this with the

   ```
   # service httpd add
   ```

   command.
3. We don't want to reboot, so start it manually:

   ```
   # service httpd start
   Starting Apache web server (no SSL)                                    [ OK ]
   ```

You have finished if you don't need SSL support.

17.4.2 Setting up SSL support for Apache

1. Creating the certifications:

```
# cd /etc/httpd/conf/
# sh mkcert.sh

Signature Algorithm ((R)SA or (D)SA) [R]:

     Here we can accept the default RSA signature algorithm first. Then
     we have to fill out some fields. There are quite a few fields but
     you can leave most of them blank. If you enter '.', the field will
     be left blank.
       1) Country Name (2 letter code) [XY]:

          Give the 2-letter code of our contry (for example US)

       2) State or Province Name (full name) [Snake Desert]:

          We type our state.

       3) Locality Name (eg, city) [Snake Town]:

         The name of our city.

       4) Organization Name (eg, company) [Snake Oil, Ltd]:

          Our organization's name.

       5) Organizational Unit Name (eg, section) [Webserver Team]:

          Our section's name.

       6) Common Name (eg, FQDN) [www.snakeoil.com]:

          Important: Give a real address here, otherwise you'll get
          warnings in your browser!

       7) Email Address (eg, 'name@FQDN') ['www@snakeoil.com']:

          I usually give the email address of the webmaster here.
          (webmaster@domain.com)

       8) Certificate Validity (days) [365]:

          In most cases, one year will be good.

          Then, we should choose the version of our certificate:

          Certificate Version (1 or 3) [3]:

          The default 3 will be good, so just hit enter. In the next
          step we can  encrypt our private key:

          Encrypt the private key now? [Y/n]:

          The keys will not be readable by users, so we can leave this
          step out.
```

So the following files are created:

```
/etc/httpd/conf/ssl.key/server.key (keep this file private!)
```

```
/etc/httpd/conf/ssl.crt/server.crt
/etc/httpd/conf/ssl.csr/server.csr
```

2. Enable SSL in */etc/httpd/conf/httpd.conf*: Open the file with your favorite editor, and search the followings at about line 1040:

```
# Uncomment this if you want SSL support!
#<IfModule mod_ssl.c>
#       Include /etc/httpd/conf/ssl.conf
#</IfModule>
```

   Uncomment them.

3. Now we should restart Apache:

```
# service httpd restart
```

4. Then we can check if the task was successful:

```
$ elinks https://localhost/
```

   This should show the default homapage, received via SSL :)

17.4.3 Self-signed Apache certificate

This must be done as root.

```
# openssl genrsa -des3 -out server.key 1024
```

Enter "foobar" twice as passphrase.

```
# openssl req -new -key server.key -out server.csr
```

Enter "foobar" when asked for passphrase, answer the questions. Leave "challenge password" "and optional company name" empty.

```
# cp server.key server.key.org
# openssl rsa -in server.key.org -out server.key
```

Enter "foobar" when asked for passphrase.

```
# openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt
# cp server.crt /etc/httpd/conf/ssl.crt/
# cp server.key /etc/httpd/conf/ssl.key/
# service httpd stop
# vi /etc/httpd/conf/httpd.conf
```

Uncomment the marked three lines around line 1044 (look for "SSL support").

```
# service httpd restart
```

Don't forget to open port 443 on your firewall, if any. (Based on How to create a self-signed SSL Certificate..., tested on frugalware-current 2007-02-14.)

17.5 asciidoc

Asciidoc has a number of configuration files under `/etc/asciidoc` and it's easy to get lost in that directory.

Regarding pdf (dblatex) generation, here are some options you can set:

- If you want to avoid the "PDF by dblatex" picture on the front page, edit `/etc/asciidoc/dblatex/asciidoc-dblatex.xsl`:

```
<xsl:param name="doc.publisher.show">0</xsl:param>
```

- If you want to avoid the "Revision History" page, add:

```
<xsl:param name="latex.output.revhistory">0</xsl:param>
```

- If you want to avoid the "Contents" page, add:

```
<xsl:param name="doc.toc.show">0</xsl:param>
```

- If you want to avoid the front page, sadly you can't do it from a configuration file, but for now you can edit `/usr/share/dblatex/latex/style/docbook.sty`. Change the `\maketitle` macro to:

```
\def\maketitle{
  \def\edhead{}
  \DBKdomitete
}
```

17.6 autojump

17.6.1 AUTOJUMP

A cd command that learns

Please read the official README or the manual.

Installation

Add the line :

```
source /etc/profile
```

to `~/.bashrc` or `~/.zshrc` if it isn't already there.

17.7 avahi

Warning
If you have rlocate installed on your system, Avahi will not run and therefore Zeroconf functionality in programs will be disabled. If you want this functionality, then please uninstall rlocate.

Also, If you are using iptables, please uncomment this line in */etc/sysconfig/firewall*:

```
#-A INPUT -p udp -m udp --dport 5353 -j ACCEPT
```

After that do not forget to restart iptables with:

```
# service firewall restart
```

17.8 b43-fwcutter

Since version 2.6.24, the bcm43xx driver is deprecated, replaced by the b43 and b43legacy modules.

The module should be loaded automatically, in case it isn't, you can load it manually:

```
# modprobe b43
```

or:

```
# modprobe b43legacy
```

You must bring the device up with ifconfig before doing any other configuration steps.

```
# ifconfig ethX up
```

Since the channel must be set manually, first do a scan:

```
# iwlist ethX scan
```

Then you can set it:

```
# iwconfig ethX channel Y
```

Finally set your essid:

```
# iwconfig ethX essid "myessid"
```

Ready!

17.9 cairo-clock

Cairo-Clock requires the Composite option to be enabled in your Xorg configuration. To enable it, add the following lines to */etc/X11/xorg.conf*:

```
Section "Extensions"
  Option "Composite" "Enable"
EndSection
```

17.10 ccache

After you installed `ccache`, it won't be enabled by default.

First, you need to determine who is allowed to use `ccache`. You have to add each user to the `ccache` group. If you want to allow using `ccache` from chrooted builds, then you need to add the `fst` user:

```
# usermod -a -G ccache fst
```

Second, you need to somehow let the build system to use `ccache`, and not the compiler directly. If you use `makepkg`, this is enabled by default (you can disable it with the `-B` option). If you build manually, then you are on your own, though usually there are two ways to do so:

- Tell the configure script to use a different compiler:

```
$ CC=/usr/bin/ccache ./configure
```

- Modify path to use the fake compiler provided by `ccache`:

```
export PATH=/usr/lib/ccache/bin:$PATH
```

17.11 cpupower

Configure your hardware specific options under /etc/sysconfig/cpupower. See the man pages for cpupower-frequency-set and cpupower-set for more information. When you are finished configuring, use this command as root to enable it at boot time:

```
systemctl enable cpupower.service
```

17.12 cryptsetup-luks

Follow these steps to when using `cryptsetup-luks`:

17.12.1 Creating

```
# cryptsetup luksFormat /dev/partition
# cryptsetup luksOpen /dev/partition label
# mke2fs -j /dev/mapper/label
# mount /dev/mapper/label /mnt/label
```

17.12.2 Mounting

Of course later you don't have to use `luksFormat` and `mke2fs`:

```
# cryptsetup luksOpen /dev/partition label
# mount /dev/mapper/label /mnt/label
```

17.12.3 Umounting

```
# umount /mnt/label
# cryptsetup luksClose label
```

17.12.4 Encrypting your home partition

Note
You have need to install the `sharutils` package to do the followings!

- List these modules in `/etc/sysconfig/modules`:

```
aes
aes-i586
sha256
dm-crypt
```

- Move all data from `/home` to a secure place (in this example `/media/sda1/home`)

```
# cp -arvx /home /media/sda1/
```

- Umount `/home` (in this example `/dev/hda6`) and fill it with random numbers:

```
# umount /home
# dd if=/dev/urandom of=/dev/hda6
```

- Create the encrypted partition:

```
# cryptsetup -y luksFormat /dev/hda6
```

Here we will be asked for a password which will be necessary to access `/home` at boot time.

- Open the encrypted partition and create its file system (`ext3` in this example):

```
# cryptsetup luksOpen /dev/hda6 home
# mkfs.ext3 /dev/mapper/home
```

- Mount the home partition and copy the contents of original home:

```
# mount /dev/mapper/home /home
# cp -arvx /media/sda1/home /home
```

- Edit the home related line in `/etc/fstab`:

```
/dev/mapper/home        /home   ext3    noatime 0       0
```

- Create `/etc/rc.d/rc.crypt` script with the following content:

```
#!/bin/sh

/usr/sbin/cryptsetup luksOpen /dev/hda6 home
/bin/mount /dev/mapper/home /home
```

- Enable it:

```
# ln -s /etc/rc.d/rc.crypt /etc/rc.d/rcS.d/S15rc.crypt
```

You have to delay the splash screen, so that you can type your password before the splash appears:

```
# mv /etc/rc.d/rcS.d/S03rc.splash /etc/rc.d/rcS.d/S15rc.splash
```

(It will ask the password between the lvm and the splash service.)

Now the system can be restarted and the password will be asked to access home partition boot-time.

Note
The English keyboard map will be used at that point of the boot process.

17.13 cwiid

17.13.1 Module loading

To use your wiimote you have to load module **uninput** with:

```
# modprobe uninput
```

To load this module at every start-up, just add **uninput** in */etc/sysconfig/modules* file.

17.14 cyrus-sasl

17.14.1 Configuring

This mini-howto helps you to install the saslauthd server using postfix which will authenticate using users and passwords from `/etc/{passwd,shadow}`.

First install the necessary packages:

```
# pacman-g2 -S postfix saslauthd
```

Enable sasl in postfix's config by appending the following lines to `/etc/postfix/main.cf`:

```
smtpd_sasl_auth_enable = yes
smtpd_sasl_local_domain = $myhostname
smtpd_sasl_security_options = noanonymous
```

You may want to append

```
broken_sasl_auth_clients = yes
```

as well.

Put the following lines to `/usr/lib/sasl2/smtpd.conf`:

```
pwcheck_method: saslauthd
mech_list: PLAIN LOGIN
```

Edit `/etc/sysconfig/saslauthd` by changing the following lines:

```
options=""
```

to

```
options="-a shadow"
```

Now you can start saslauthd by

```
service saslauthd start
```

as well as enabled in by default on startup:

```
service saslauthd add
```

Issue `id postfix` and see if the `daemon` group is listed. If not, then add `postfix` to the `daemon` group:

```
usermod -G daemon postfix
```

Finally restart postfix:

```
service postfix restart
```

Compeleted!

17.14.2 Verifying

We test it using telnet. We need perl to generate the string for the SASL authentication:

```
$ perl -MMIME::Base64 -e 'print encode_base64("vmiklos\0vmiklos\0secret");'
dm1pa2xvcwB2bWlrbG9zAHNlY3JldA==
```

Then use telnet:

```
$ telnet host.com 25
Trying ip...
Connected to host.com.
Escape character is '^]'.
220 host.com ESMTP Postfix
ehlo my.dhcp
250-host.com
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-AUTH LOGIN PLAIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
AUTH PLAIN dm1pa2xvcwB2bWlrbG9zAHNlY3JldA==
235 2.0.0 Authentication successful
quit
221 2.0.0 Bye
Connection closed by foreign host.
```

17.15 dante

17.15.1 Configuration

In most cases you have a socks server (you can create one easily using ssh, see the documentation of the openssh package), and you want to route all traffic through it. Here is the config you need:

```
route {
        from: 0.0.0.0/0 to: 0.0.0.0/0 via: 127.0.0.1 port = 8080
        proxyprotocol: socks_v4
}
```

17.15.2 Testing it

Try for example:

```
$ socksify irssi
```

When you connect to a server, others will see that you're connecting from the server, not from your own host.

17.16 ddclient

Please configure */etc/ddclient/ddclient.conf* before running ddclient!

Samples for common configurations can be found in: /usr/share/doc/ddclient-$package_version/sample*

Additional details and instructions can be found in: /usr/share/doc/ddclient-$package_version/README

Once you have finished configuring the *ddclient.conf* file, you can start ddclient as a daemon by running as root, the following command:

```
# service ddclient start
```

17.17 dhcp

If you are in trouble setting up your dhclient, use the following options. These are quite good defaults:

```
request subnet-mask, broadcast-address, time-offset, \
        routers, domain-name, domain-name-servers, \
        host-name, netbios-name-servers, netbios-scope;
timeout 20;
script "/sbin/dhclient-script";
```

17.18 drupal6

To be able to use this package as intended, you will have to:

- set up apache to access `/var/www/drupal6` from the web the way you like;
- install and set up your favourite SQL database (mysql or postgresql; this package DOES NOT depend on any of them);
- create and/or grant access to a mysql or postgresql database;
- set up your drupal installation itself by entering the correct credentials at the install screen to be able to reach the above-mentioned database.

17.19 drupal7

To be able to use this package as intended, you will have to:

- set up apache to access `/var/www/drupal7` from the web the way you like;
- install and set up your favourite SQL database (mysql, postgresql or sqlite; this package DOES NOT depend on any of them);
- create and/or grant access to a mysql, postgresql or sqlite database;
- set up your drupal installation itself by entering the correct credentials at the install screen to be able to reach the above-mentioned database.

17.20 dspam

To populate the DSPAM database, you need to follow several steps.

1. First create a database. Login to the mysql command prompt.

   ```
   $ mysql -u root -p
   mysql> CREATE database dspam;
   ```

2. Next, you need to create a dspam user. At the same MySQL prompt:

   ```
   mysql> GRANT ALL PRIVILEGES ON dspam.* TO dspam@'localhost' IDENTIFIED BY 'passwd';
   ```

   Replacing passwd with your chosen password.

3. Optimizing the datebase:

   If you want a space optimized db do:

```
$ mysql -u dspam dspam -p < /var/lib/dspam/mysql/mysql_objects-space.sql
```

If you want a speed optimized db do:

```
$ mysql -u dspam dspam -p < /var/lib/dspam/mysql/mysql_objects-speed.sql
```

Enter the password you set in the previous step, and the database should be populated.

4. Remember to edit /etc/dspam/dspam.conf accordenly

If you want to use the postgresql, sqlite3 or Berekely DB4 backends you can find instructions in the dspam documentation.

17.21 eaccelerator

17.21.1 Setting up eaccelerator

In order to use eAccelerator, you must add the following lines to your */etc/php.ini* file:

```
extension="/usr/lib/php/extensions/no-debug-non-zts-20090626/eaccelerator.so"
eaccelerator.shm_size="16"
eaccelerator.cache_dir="/tmp/eaccelerator"
eaccelerator.enable="1"
eaccelerator.optimizer="1"
eaccelerator.check_mtime="1"
eaccelerator.debug="0"
eaccelerator.filter=""
eaccelerator.shm_max="0"
eaccelerator.shm_ttl="0"
eaccelerator.shm_prune_period="0"
eaccelerator.shm_only="0"
eaccelerator.compress="1"
eaccelerator.compress_level="9"
```

Do not forget to create the cache directory as well:

```
mkdir /tmp/eaccelerator
chmod 0777 /tmp/eaccelerator
```

17.21.2 Configuration Options:

```
eaccelerator.shm_size
    The amount of shared memory (in megabytes) that eAccelerator will use.
    "0" means OS default. Default value is "0".

eaccelerator.cache_dir
    The directory that is used for disk cache. eAccelerator stores precompiled
    code, session data, content and user entries  here. The same data  can  be
    stored in shared memory also (for more quick access). Default value is
    "/tmp/eaccelerator".

eaccelerator.enable
    Enables or disables eAccelerator. Should be "1" for enabling  or  "0"  for
    disabling. Default value is "1".

eaccelerator.optimizer
    Enables or disables internal peephole optimizer which may  speed  up  code
    execution. Should be "1" for enabling or "0" for disabling. Default  value
    is "1".
```

```
eaccelerator.debug
    Enables or disables debug logging. Should be "1" for enabling or  "0"  for
    disabling. Default value is "0".

eaccelerator.check_mtime
    Enables or disables PHP file modification checking .  Should  be  "1"  for
    enabling or "0" for disabling. You should set it to "1"  if  you  want  to
    recompile PHP files after modification. Default value is "1".

eaccelerator.filter
    Determine which PHP files must be cached. You may specify  the  number  of
    patterns (for example "*.php *.phtml") which specifies to cache or not  to
    cache. If pattern starts with the character "!", it means to ignore  files
    which are matched by the following pattern. Default value is "" that means
    all PHP scripts will be cached.

eaccelerator.shm_max
    Disables putting large values into shared memory by " eaccelerator_put() "
    function. It indicates the largest allowed size in bytes (10240, 10K, 1M).
    The "0" disables the limit. Default value is "0".

eaccelerator.shm_ttl
    When eaccelerator fails to get shared memory for new script it removes all
    scripts which were not accessed  at  last "shm_ttl"  seconds  from  shared
    memory. Default value is "0" that means -  don't  remove  any  files  from
    shared memory.

eaccelerator.shm_prune_period
    When eaccelerator fails to get shared memory for new script  it  tryes  to
    remove  old  script   if   the   previous   try   was   made   more   then
    "shm_prune_period" seconds ago. Default value is "0" that  means  -  don't
    try to remove any files from shared memory.

eaccelerator.shm_only
    Enables or disables caching of compiled scripts on disk. It has  no  effect
    on session data and content caching. Default value is "0" that means -  use
    disk and shared memory for caching.

eaccelerator.compress
    Enables or disables cached content compression. Default value is  "1"  that
    means enable compression.

eaccelerator.compress_level
    Compression level used for content caching.  Default value is "9" which  is
    the maximum value

eaccelerator.keys
eaccelerator.sessions
eaccelerator.content
    Determine where keys, session data and content will be cached. The possible
    values are:
    "shm_and_disk" - cache data in shared memory and on disk (default value)
    "shm"          - cache data in shared memory or on disk if shared memory
                     is full or data size greater then "eaccelerator.shm_max"
    "shm_only"     - cache data in shared memory
    "disk_only"    - cache data on disk
    "none"         - don't cache data


eAccelerator API:
```

```
eaccelerator_put($key, $value, $ttl=0)
  puts the $value into shard memory for $ttl seconds.

eaccelerator_get($key)
  returns the value from shared memory which was stored by  eaccelerator_put()
  or null if it is not exists or was expired.

eaccelerator_rm($key)
  removres the $key from shared memory

eaccelerator_gc()
  removes all expired keys from shared memory

eaccelerator_lock($lock)
  creates a lock with specified name. The lock can  be  released  by  function
  eaccelerator_unlock() or automatic on the end of request.
  For Example:
  <?php
    eaccelerator_lock("count");
    eaccelerator_put("count",eaccelerator_get("count")+1));
  ?>

eaccelerator_unlock($lock)
  release lock with specified name

eaccelerator_set_session_handlers()
  install the eaccelerator session handlers.
  Since PHP 4.2.0 you can install eaccelerator session handlers
  in "php.ini" by "session.save_handler=eaccelerator".

eaccelerator_cache_output($key, $eval_code, $ttl=0)
  caches the output of $eval_code in shared memory for $ttl seconds.
  Output can be removed from cache by calling mmcach_rm() with the same $key.
  For Example:
  <?php eaccelerator_cache_output('test', 'echo time(); phpinfo();', 30); ?>

eaccelerator_cache_result($key, $eval_code, $ttl=0)
  caches the result of $eval_code in shared memory for $ttl seconds.
  Result can be removed from cache by calling mmcach_rm() with the same $key.
  For Example:
  <?php eaccelerator_cache_output('test', 'time()." Hello";', 30); ?>

eaccelerator_cache_page($key, $ttl=0)
  caches the full page for $ttl seconds.
  For Example:
  <?php
    eaccelerator_cache_page($_SERVER['PHP_SELF'].'?GET='.serialize($_GET),30);
    echo time();
    phpinfo();
  ?>

eaccelerator_rm_page($key)
  removes the page which was cached by eaccelerator_cache_page() with the same
  $key from cache

eaccelerator_encode($filename)
  returns the encoded bytecode of compiled file $filename

eaccelerator_load($code)
  loads script which was encoded by eaccelerator_encode()
```

17.22 ejabberd

17.22.1 Creating your SSL keys

Generate Key Pair:

```
# cd /etc/ejabberd
# openssl req -new -x509 -newkey rsa:1024 -days 3650 -keyout privkey.pem -out server.pem
```

Note
You should enter your domain name as the Common Name for your certificate.

Remove pass parse:

```
# openssl rsa -in privkey.pem -out privkey.pem
```

Combine the Private and Public Key:

```
# cat privkey.pem >> server.pem
```

Delete Private Key:

```
# rm privkey.pem
```

Set permissions:

```
# chown root:ejabberd server.pem
# chmod 640 server.pem
```

Finally update the config file:

- Change the `./ssl.pem` string to `/etc/ejabberd/server.pem`.
- Change `starttls` to `tls` in the `listen` section if you want to force users to use SSL.

17.22.2 Creating an administrator

Register an account on your `ejabberd` deployment. An account can be created using a jabber client like `pidgin`.

Add the following lines to you config:

```
{acl, admins, {user, "admin", "example.org"}}.
{access, configure, [{allow, admins}]}.
```

This will promote the account created in the previous step to an account with administrator rights.

17.22.3 Testing

Add the following line to your `/etc/sysconfig/firewall`, for example after `mysql`:

```
# ejabberd
-A INPUT -p tcp -m tcp --dport 5222 -j ACCEPT
```

Now you should be able to connect to `ejabberd` remotely. Start your favourite jabber client on a remote machine (ie. pidgin) and register another account. You should be able to talk to the admin now and vica versa.

For more info, please read the Installation and Operation Guide, which can be found at `/usr/share/doc/ejabberd-*/guide.html`.

17.23 enemy-territory

Evenbalance, developer of Punkbuster dropped support for Wolfenstein Enemy Territory (ET). Also the Punkbusterinstaller isn`t able to install the neccessary files for Enemy Territory. So if you got disconnected from servers and getting some #20004 errors, you can run *et-pbupdate* instead of *pbweb*. You can read more: http://etkey.org/

17.24 fbterm

To configure fbterm, please edit `/etc/fbtermrc`.

17.25 fuse

Fuse is a virtual filesystem "helper" which makes possible to mount unusual things as a filesystem. It is achieved by using a simple program, which runs in user space, to provide data that can be represented by the fuse kernel module as a filesystem. The interpreter program is a less complex one than a kernel-space module, which is much harder to write. In Frugalware, regular users of a given box can mount filesystems by fuse. First as root, let's install the tools needed:

```
# pacman-g2 -S fuse
```

Now, having the base of fuse, we need to install the programs for each specific filesystem type. To get a hint on what is available, you can issue the following command:

```
$ pacman-g2 -Ss fuse
```

The two most used (ftp, ssh) plugins can be installed by running the following command. Beware, the ftp fs is a perl module, and it seems a bit memory hungry / buggy / slow so therefore it might be replaced by CurlFtpFS in the future.

```
# pacman-g2 -S fuseftp sshfs-fuse
```

Then, you can mount a remote dir with sftp access as a regular user doing:

```
$ /sbin/mount.fuse  sshfs#YOURUSERNAME@SERVER:/REMOTEDIR /LOCALDIR -o rw,OTHEROPTIONS
```

You can also unmount it as a regular user doing:

```
$ fusermount -u /LOCALDIR
```

17.26 fw32

17.26.1 Initial setup

Edit /etc/fw32/pacman-g2.conf if you want to change the mirror used, or other options used for pacman-g2.

Commands to use (with sudo or root shell):

```
fw32-create
systemctl enable fw32.service (required for boot-time fw32 root mounting)
```

17.26.2 Upgrading chroot

This needs to be done when packages become out of date. Command to use (with sudo or root shell):

```
fw32-upgrade
```

Warning
Should not be used while someone is using the chroot.

17.26.3 Installing packages or groups to chroot

Command to use (with sudo or root shell):

```
fw32-install <packages and/or groups>
```

Warning
Should not be used while someone is using the chroot.

17.26.4 Removing packages or groups from chroot

Command to use (with sudo or root shell):

```
fw32-remove <packages>
```

Warning
Should not be used while someone is using the chroot.

17.26.5 Installing local FPM package to chroot

Command to use (with sudo or root shell):

```
fw32-install-package <FPM packages>
```

Warning
Should not be used while someone is using the chroot.

17.26.6 Installing nobuild package to chroot

Command to use (with sudo or root shell):

```
fw32-merge <package>
```

Warning
Should not be used while someone is using the chroot.

17.26.7 Cleaning chroot cache

Command to use (with sudo or root shell):

```
fw32-clean
```

Warning
Should not be used while someone is using the chroot.

17.26.8 Deleting chroot

Command to use (with sudo or root shell):

```
fw32-delete
```

Warning
Should not be used while someone is using the chroot.

17.26.9 Removing fw32

Command to use (with sudo or root shell):

```
fw32-delete
systemctl disable fw32.service (only needed if you enabled this at setup time)
rm -f /var/cache/pacman-g2/pkg/*i686.fpm (only needed if you want to delete the fpm cache)
pacman-g2 -R fw32
```

Warning
Should not be used while someone is using the chroot.

17.26.10 Running a command within the chroot

Commands run will have the permissions of the user.

To get a shell:

```
fw32-run
```

To run a specific command:

```
fw32-run <command> [<arguments>]
```

17.26.11 Commands

- fw32-clean: Clean the cache of old packages.

Warning
Should not be used while someone is using the chroot.

- fw32-create: Create the initial chroot.
- fw32-delete: Delete the chroot, ensuring everything is umounted.

Warning
Should not be used while someone is using the chroot.

- fw32-install: Install all packages and groups specified to the chroot.

Warning
Should not be used while someone is using the chroot.

- fw32-install-package: Install all i686 FPMs specified to chroot.

Warning
Should not be used while someone is using the chroot.

- fw32-merge: Install a nobuild package to chroot.

Warning
Should not be used while someone is using the chroot.

- fw32-mount-all: Manually mount the chroot base directories.
- fw32-run: Run a command within the chroot. If no command is specified, an attempt is made to execute the user's shell.
- fw32-umount-all: Manually umount all the directories in the chroot.

Warning
Should not be used while someone is using the chroot.

- fw32-remove: Remove all packages or groups specified from the chroot.

Warning
Should not be used while someone is using the chroot.

- fw32-upgrade: Performs a system upgrade inside the chroot.

Warning
Should not be used while someone is using the chroot.

17.26.12 building i686 packages

Use the command `fw32-makepkg` as root, in the same way you would use regular makepkg. It will transparently wrap your build into a i686 chroot to produce a i686 package.

17.26.13 nobuild packages

Some `nobuild` packages (like Skype) are available on x86_64, even if upstream provides an i686 binary only. In that case the package has to be installed inside the i686 chroot and on the host system as well: the host package will contain a desktop file and an icon only to invoke the chrooted package. See the fw32-merge command for details on how to install the i686 version.

17.27 gammu

17.27.1 Configuring

You need to create your ~/.gammurc:

```
[gammu]
port = /dev/ttyUSB0
connection = fbus
```

Replace `/dev/ttyUSB0` with your serial port device and `fbus` with the appropriate protocol name if you are not a Nokia user. Check if you have write access to the device, you need to be a member of the `uucp` group.

Once you think you're done, check your setup:

```
$ gnokii --identify
```

It should print your `IMEI` number so that you'll be able to check if `gammu` really found your phone or there is a problem.

17.27.2 Creating a backup

You probably use `gammu` to make a backup of your phone.

This involves two steps:

- Backing up your SMSes

```
$ gammu --backupsms backupsms.txt
```

- The rest of your phone.

```
$ gammu --backup backup.txt
```

You may find an alternative format more human-readable for SMSes:

```
$ gammu --geteachsms > eachsms.txt
```

See the manual page for more tricks!

17.28 gif2png

If you want to use *web2png*, you must install *python*.

pacman-g2 -S python == git

17.28.1 gitweb

If you want to set up a web interface for your git repositories, then:

- install the `gitweb` package
- edit `/etc/gitweb.conf` so that `$projectroot` will point to the repository directory
- restart `apache` so that the `gitweb` configuration will be included.

17.29 gnome-bluetooth

For have a full bluetooth support with gnome install obex-data-server # pacman-g2 -S obex-data-server

17.30 grub2

It is no longer acceptable to edit your grub configuration manually since upgrading to grub2. Instead, it is advised to insert any customizations you require in `/etc/sysconfig/grub-config` and `/etc/sysconfig/grub-custom`.

17.31 help2man

The most common usage of this applications is something like this:

```
$ help2man -n "<oneliner description>" -S Frugalware -N ./<binary> |sed 's/\\(co/(c)/' >< ↩
    binary>.1
```

17.32 horde-webmail

This app does not have any webserver, SQL server nor IMAP server in its depends, which is intentional. Anyway, if you plan to use it, you should set up a webserver and an IMAP server. The SQL server is optional, but it's the most easiest-to-use preferences container.

Additionally this app is not configured in any way: there are far too many customizable settings, so the packager cannot know how to set them for your particular needs. Installation instructions can be found in the INSTALL file.

17.33 hostapd

Configuration examples can be found in /etc/hostapd. You must edit the following files located in /etc/hostapd to configure hostapd:

hostapd.allow hostapd.conf hostapd.deny

17.34 icewm

I have included a custom shell script called icewm-menus, for use with the icewm menu file. An example menus file is also include at /usr/share/icewm/menus. It uses standard shell syntax, so you can easily use shell variables, etc, to create dynamic menus in icewm through my script and the usage of your local $HOME/.icewm/menus file. To use it, use the following syntax in your menu file: menuprog "(folder name)" (icon name) icewm-menus (menu switch to use) If setup correctly, you'll wind up with menus generated by the output of the shell script. Have fun configuring icewm.

17.35 k3b

If you want to rip a video DVD, install the transcode package as well.

17.36 kbstick

If you do not know the keycodes for the keys you wish to remap the joystick events to, then please install the xev program. It will help you to identify them. Moving on, the /etc/kbstick.conf is the system level configuration file the shell script reads from if the user does not have a .kbstickrc in their home directory. Syntax is the same in both cases, and the configuration file has some comments to give you an idea of what each variable does. I have set the default up/down/left/right key mappings to what my laptop uses for them and the buttons will have to be manually defined to their proper keycodes. If you need any further help, please email the maintainer of this package.

17.37 kexec-tools

Warning
kexec works just like *reboot*, so please save your data before using it!

Loading the new kernel:

```
# kexec -l /boot/vmlinuz-2.6.18-fw1 --append="ro root=/dev/hda3 quiet resume=/dev/hda2"
```

Booting it:

```
# kexec -e
```

17.38 keychain

First of all, we have to install package called keychain. (`pacman-g2 -S keychain`)

In the next step we have to create a new key. A key stands from two parts, a public and a private part. It means two different files in your `~/.ssh/` directory.

Your key is generated by a program called ssh-keygen. It's a part of openssh package. Run ssh-keygen -t dsa! You'll see something like this:

```
voroskoi@kavics~$ ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/voroskoi/.ssh/id_dsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/voroskoi/.ssh/id_dsa.
Your public key has been saved in /home/voroskoi/.ssh/id_dsa.pub.
The key fingerprint is:
ac:47:93:29:d2:c4:e1:85:47:5c:c1:36:93:74:e9:08 voroskoi@kavics
```

It'll generate for us the two parts of the key. The program asks where do you want to save the keys, it's good to simply push an enter. After that You have to type in the passphrase of the key two times. It's *really* important to chose a hard passphrase. It should contain lower-/uppercase characters, digits, possibly special characters too. The length must be at least 10 characters! We have to type in this passphrase only once after every restart we shouldn't choose an easy one.

If everything works fine, then we have an id_dsa and an id_dsa.pub file in our `~/.ssh/` directory.

```
voroskoi@kavics~/.ssh $ ls -la
drwx------   2 voroskoi users   5 2005-04-13 13:39 ./
drwx--x--x  38 voroskoi users  67 2005-04-13 13:24 ../
-rw-------   1 voroskoi users 736 2005-03-01 21:25 id_dsa
-rw-r--r--   1 voroskoi users 605 2005-04-11 04:18 id_dsa.pub
-rw-r--r--   1 voroskoi users 230 2005-04-11 04:26 known_hosts
```

Now, we would like to use our newly generated key. We have to do the following:

```
$ scp ~/.ssh/id_dsa.pub username@remote_machine:
$ ssh username@remote_machine
$ cat id_dsa.pub >> ~/.ssh/authorized_keys
$ rm id_dsa.pub
$ exit
```

Good to know, that this time(I mean when we run scp and ssh commands) we can't use our key's passphrase, so we have to use our password on the remore_machine. If it's done without any mistake on next login the remote_machine will ask for our key's passphrase.

And here comes keychain. In openssh package there is a program called ssh-agent. You can store keys in ssh-agent. Keychain just makes easier using of ssh-agent and adds some new features.

This time i assume that we use bash. If we would like to use keychain with an other shell, then we can use man keychain:-) So, let's take out favourite editor and add the following lines to `~/.bash_profile` file:

```
keychain -q id_dsa
[ -f $HOME/.keychain/$HOSTNAME-sh ] && source $HOME/.keychain/$HOSTNAME-sh
```

17.39 ksplice

ksplice is handy in case there is a serious security fix and you don't want or can't afford rebooting your system immediately.

Let's pick an example, the kernel-2.6.28-6anacreon3 update, which added CVE-2009-2692.patch.

First update FST so that you will have the patch:

```
# repoman upd
```

Now create a working dir:

```
$ cp -a /usr/src/linux/ ~/linux-source
$ cd ~/linux-source
$ mkdir ksplice
$ cp /boot/config ksplice/.config
```

```
$ cp /boot/System.map ksplice/
$ ln -s ~/linux-source ksplice/build
$ cp /var/fst/stable/source/base/kernel/CVE-2009-2692.patch .
```

Now create the ksplice update:

```
$ ksplice-create --patch=CVE-2009-2692.patch ~/linux-source
```

Then apply it:

```
# ksplice-apply ksplice-st4dt4bg.tar.gz
```

To view all applies updates, or a specific one:

```
# ksplice-view
# ksplice-view --id=st4dt4bg
```

To revert one:

```
# ksplice-undo st4dt4bg
```

17.40 kvpnc

Howto setup KVpnc for use without root password - sudo

1. Install sudo
2. Edit */etc/sudoers*: add an command alias

```
# Cmnd alias specification
Cmnd_Alias KVPNC = /usr/bin/kvpnc

# User privilege specification
ALL ALL=NOPASSWD:KVPNC
```

Warning
Do it gently! (As always, when you edit */etc/sudoers*.)

17.41 lastfmsubmitd

17.41.1 Configuring Lastfmsubmitd

Change your LastFM username and password in `/etc/lastfmsubmitd.conf` and the MPD server settings in `/etc/lastmp.conf` before starting the LastFM submit daemon.

17.41.2 Starting the daemon(s)

After configuring `lastfmsubmitd`, you should run the following commands to start the daemons:

```
# systemctl start lastfmsubmitd.service
# systemctl start lastmp.service
```

17.42 lesspipe

For syntax highlighting support in `less` via the `lesspipe` wrapper, you must install the `source-highlight` package.

17.43 lilo

So, you feel like using `lilo`, do you? Well, here you will find instructions for configuring `lilo` to work with Frugalware. Some things to keep in mind:

1. `lilo` must be rerun every time you upgrade the kernel
2. `lilo` must also be rerun if you change configuration for it to take effect
3. only `lilo` or grub can be installed to your boot sector at the same time, however they do not conflict while simply residing on your system

You will find an example `lilo.conf` in `/etc/lilo.conf` already. You will need to tweak it in order for it to match your system's booting setup. The default structure is designed to reflect the most common setup I know of, but may still require a lot of modifications. For more information on `lilo`, please refer to `man lilo` and `man lilo.conf`.

17.44 lineakd

After installing lineakd, make sure you create a configuration file before starting it.

Example configuration files are located in /usr/share/doc/lineakd-*/.

Don't forget to copy the configuration file to /etc/lineakd after you create it.

You can then start the lineak daemon by running the following command:

```
$ lineakd
```

17.45 lirc

After installing lirc you need to take the following steps:

1. Find a lird.conf for your remote control on remotes You can also take a look on */usr/share/remotes* directory if you do not have an internet connection. If you do not find your remote controller, try `irrecord myremote` command.
2. Copy your lircd.conf to */etc/* directory as root.
3. Add evdev to */etc/sysconfig/modules*.
4. Load the module with `modprobe evdev`.
5. Edit */etc/sysconfig/lirc* if necessary.

   ```
   $ cat /proc/bus/input/devices | grep -e N -e H
   ```

   will show you the event# you should use. (Default is 2.)
6. Start lircd and lircmd with `sudo service lirc start`.

17.46 lmsensors

lmsensors is a hardware monitoring tool which is able to read thermal and voltage values and fan speeds from the sensor chips of your motherboard. Before running sensors you have to run sensors-detect as root to initialize them. It will autodetect your hardware and define which kernel modules you need to get it working properly, and tell you how to autoload them during boot.

So if you want to use lmsensors try to run

```
sensors-detect
```

and say YES at end of sensors-detect to write `/etc/sysconfig/lm_sensors`.

17.47 lvm2

17.47.1 Creating

Here is a mini-HOWTO, a longer one is available here.

First if you are on a setup cd, you need to

```
modprobe dm-mod
```

and

```
vgchange -a y
```

The first loads the device-mapper support for the kernel, the later enables the existing volume groups. This is automatically done for you on an installed Frugalware system.

You need to decide what physical partitions to use for LVM. In this mini-HOWTO `/` is `/dev/hda1` and we create a big `/home` partition using `/dev/hda2` and `/dev/hdc1`.

Let's initialize them for use by LVM:

```
pvcreate /dev/hda2 /dev/hdc1
```

Create a volume group titled `vg`:

```
vgcreate vg /dev/hda2
```

Extend it with `/dev/hdb1`:

```
vgextend vg /dev/hdc1
```

Then we can create a logical volume with a size of 400G titled `home`:

```
lvcreate -L400G -nhome vg
```

Create a filesystem on it as usual, ie. for ext3:

```
mke2fs -j /dev/vg/home
```

And now the only task is to mount it as usual, ie:

```
mount /dev/vg/home /mnt/target/home
```

17.47.2 Extending

You already saw how to extend a volume group. Extending a logical volume is a bit more complex, but still easy.

If you use ext3:

```
umount /mnt/target/home
lvextend -L+900M /dev/vg/home
resize2fs /dev/vg/home
mount /dev/vg/home /mnt/target/home
```

Note
According to the manpage of resize2fs, it would have support resizing without umounting, but this does not seem to work.

If you use reiserfs:

```
lvextend -L+900M /dev/vg/home
resize_reiserfs /dev/vg/home
```

17.47.3 Removing

To remove a logical volume:

```
lvremove /dev/vg/home
```

To remove a physical volume from a volume group:

```
vgreduce vg /dev/hdc1
```

To remove a volume group:

```
vgremove vg
```

That's it.

17.48 mailman

There is no any kind of http server in mailman's depends. It's because they are not needed to get a working mailman. Of cource if you want to provide archives and so don't forget to install a http server.

17.49 man-db

If you like coloured man-pages then you can enable that feature by issuing

```
# chmod +x /etc/profile.d/man-colors.sh
```

It is handled as a configuration file, so feel free to edit the colors in that file if you want.

17.50 mantis

You have to GRANT some privileges (at least for the operating user) to be able to use this package, as the installer does not GRANT them. The operating user requires ALTER, SELECT, INSERT, UPDATE and even DELETE privileges, regardless that the latter is not mentioned by upstream. For installation, INDEX, CREATE, DELETE, and DROP privileges are also required - this can be carried out if you provide the (MySQL) superuser's credentials to the installer.

Do not forget to `rm -rf /var/www/mantis/admin` after a successful install to prevent hijacking your bugtracker, and change the default administrator's password.

17.51 mediawiki

After installing this package, please run */usr/bin/mediawikisetup* as root to setup MediaWiki

17.52 mod_mono

For enable mod_mono module apache don't forget to define the User/Group directives into /etc/httpd/conf/httpd.conf. For test the configuration of mod_mono into /etc/httpd/conf/httpd.conf : #mono settings Alias /demo /usr/lib/xsp/test MonoApplications "/demo:/usr/lib/xsp/test" MonoServerPath /usr/lib/mono/2.0/mod-mono-server2.exe <Directory /usr/lib/xsp/test> SetHandler mono </Directory> and check the result : http://localhost/demo/

17.53 monit

You may want to forge a config file for yourself as */etc/monit/monitrc* to be able to properly use Monit. Consult the online docs for details:

http://mmonit.com/monit/documentation/monit.html

After doing so you should issue a `systemctl enable monit.service` command to make use of this service.

17.54 motion

You should edit the settings: videodevice, input, norm, frequency, width, height and target_dir in */etc/motion.conf*

17.55 munin

From munin-1.2.5-2 we no longer use a random uid/gid, but dedicated ones. Because of this munin service will not start if you have it installed before, so you have to correct this by issuing these commands:

```
groupmod -g 47 munin
usermod -u 47 -g 47 munin
chown -R munin:munin /var/lib/munin
chown -R munin:munin /var/www/html/munin
chown -R munin:munin /var/log/munin
chown -R munin:munin /var/run/munin
```

You should chown any other munin-owned stuff you may have lying around, these are only the default ones.

17.56 nss-mdns

To enable IPv4 multicast DNS lookups, append *mdns4* to the hosts line in */etc/nsswitch.conf*. Use *mdns6* for IPv6 or *mdns* for both.

17.57 openssh

17.57.1 Forwarding ports

```
# ssh -L 8000:localhost:80 server.com
```

After this you can access server.com:80 at localhost:8000 even if server.com:80 is not accessible from your machine.

17.57.2 Socks proxy

Many mobile users have the following problem: they have to use an unencrypted wireless lan and they want to access webservers which does not support https. There is an easy solution for this: you transfer data to a server in an ssh tunnel then the data can be transferred to the server unencrypted in a wired network. This is much more secure. Set up the socks proxy on localhost:8080:

```
$ ssh -D 8080 server.com
```

Then configure your webbrowser to use the proxy, for example in firefox, select `Manual proxy configuration` and then set `SOCKS Host` to localhost, `Port` to 8080.

Note
Don't forget to clear other proxy fields! (HTTP, SSL, FTP, etc.)

17.58 pawm

Copy /etc/pawm.conf to $HOME/.pawm for your own local changes. If you want icons on your desktop, add a file to your $HOME/.pawm directory that starts with "app" and append an alphanumerical phrase of your choice to it. Then, write the file structure as follows:

<icon name> <x position> <y position> <name to display> <command>

Example:

firefox.xpm 40 40 firefox firefox

Other things to remember, you can only use xpm files for this method, and it takes the files from /usr/share/pixmaps. If I knew how to change this path to a directory the user has, I would.

17.59 pdns

If you wish to use the gmysql or gpgsql backends with a local server, then follow these instructions.

For gmysql, install `mysql` package.

```
pacman-g2 -Sy mysql
```

For gpgsql, install `postgresql` package.

```
pacman-g2 -Sy postgresql
```

Now, copy `/lib/systemd/system/pdns.service` to `/etc/systemd/system/pdns.service`.

```
cp -f /lib/systemd/system/pdns.service /etc/systemd/system/pdns.service
```

Uncomment the lines appropriate for your selected backend. The comments in the file will guide you. After all this, you must still ensure the specific database backend you are wanting to use is properly configured. This means both the pdns configuration and the setup for the mysql or postgresql daemon. Refer to pdns, mysql, and/or postgresql documentation for more information.

17.60 pekwm

Be sure to make your own file at $HOME/.pekwm/autostart if you use pekwm-session to auto-launch commands when you startup. I know pekwm has a start file for this, but my method launches it **only** at the start of your session, while the method pekwm uses starts everytime you restart/start pekwm. Use it well. You can find an example below:

dbus-session --exit-with-session --sh-syntax & feh --bg-scale "$HOME/.foo/bar" &

17.61 perlpanel

I have purposely left out a few perl modules from the dependencies array, because they are not needed to run perlpanel and drag in a lot of GNOME or other stuff you may not want. Below, you will find a list of these modules and what they do. If you find errors in this documentation, then please report it and I will look into it.

perl-xmms - perlpanel plugin interface to xmms perl-gnome2-vfs - various gnome plugin interfaces for perlpanel libgnomeui - for full libglade support in perlpanel

17.62 phc-optimizer

This package contains a script for finding the optimal voltage while maintaining system stability. During the process, your system will mostly likely crash multiple times before you find the right settings. Make sure you are not running or doing anything important while using this script. Keep a backup of essential data in case of data loss.

Now, you will need to run this script as root or have sudo privileges. In addition, you need to have installed either phc-intel or phc-k8 and have compatible hardware. Run this command as root or regular user with sudo privileges, and follow the interactive prompts it gives you.

```
phc-optimizer
```

It will save the results from testing in a file called `phc_tweaked_vids` in the directory it was executed from. This process should be repeated for each VID value. After all this work, you should have your final set of VIDs.

17.63 php

You should set

```
cgi.fix_pathinfo=1
```

in */etc/php.ini* in order to use php-cgi.

17.64 php-jsmin

17.64.1 Setting up JSMin

In order to use JSMin, you must add the following lines to your */etc/php.ini* file:

```
extension="/usr/lib/php/extensions/no-debug-non-zts-20090626/jsmin.so"
```

17.65 phpbb

After installing this package, please run */usr/bin/phpbbsetup* as root to setup phpBB

After upgrading, make sure to run the database update script

17.66 pm-radeon

Before you can use this package, you must edit the configuration for it in the file `/etc/sysconfig/pm-radeon`. After you are done, run this command to enable it at startup.

```
systemctl enable pm-radeon.service
```

17.67 pootle

In most cases you want to use pootle with mysql and apache. See here on how to configure them:

- http://translate.sourceforge.net/wiki/pootle/using_mysql
- http://translate.sourceforge.net/wiki/pootle/apache

Also read these pages if you're upgrading from Pootle 1.x:

- http://translate.sourceforge.net/wiki/pootle/important_changes
- http://translate.sourceforge.net/wiki/pootle/migration

17.68 postfix

17.68.1 Using a relay host

These are the basic steps to set up Postfix to use SMTP Authentication to send mail through a relay host.

Set up a password maps file (`/etc/postfix/sasl_passwd`) as follows:

```
mail.ispserver.com    username:password
```

```
# chown root:root /etc/postfix/sasl_passwd
# chmod 600 /etc/postfix/sasl_passwd
# postmap /etc/postfix/sasl_passwd
```

Append the following lines to `/etc/postfix/main.cf`:

```
relayhost = mail.ispserver.com
smtp_sasl_auth_enable = yes
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
smtp_sasl_security_options =
```

Finally reload postfix:

```
# postfix reload
```

That should do it!

17.69 postfixadmin

This package relies on correct install of postfix's virtual tables and it needs to be configured before usage. Be sure to read upstream's `/var/www/postfixadmin/INSTALL.TXT` in order to accomplish the setup or upgrade. You should also take care of configuring apache to be able to use the web-based interface.

Should you want to make use of the autoreply (vacation) feature, install these packages as well: `perl-mail-sender`, `perl-email-valid`, `perl-mime-charset`, `perl-log-dispatch`, `perl-mime-encwords`, `perl-params-validate` and read upstream's `/var/www/postfixadmin/VIRTUAL_VACATION/INSTALL.TXT` in order to setup autoreply (vacation) properly. Don't forget to enable it in `config.inc.php` as well!

17.70 postgrey

To use postgrey, put something along the lines of

```
    smtpd_recipient_restrictions =
        ...
        reject_unauth_destination
        check_policy_service inet:127.0.0.1:60000
```

in your /etc/postfix/main.cf (postfix 2.1 or newer is required.)

17.71 pptpd

1. Preface

   I was asked to set up VPN using PPTP. A much secure way to setup it up is using IPSec, more details here. Also you could use ssh+pppd, but that's rather problematic on platforms other than Unix.

2. Setting up the server

   The big problem here is that most outdated HOWTO starts with patching your kernel and ppp. This is no longer needed!

   Requiements: You need kernel>=2.6.15 or newer (Frugalware 0.4 or higher is OK). Also you need ppp>=2.4.2.

   Also probably these are already installed on your system, let's see the new package: pptpd. Install it with the usual

   ```
   # pacman-g2 -S pptpd
   ```

   Probably this is done if you're reading this HOWTO :-)

   Here comes my */etc/pptp.conf*:

   ```
   $ grep -v '^\(#\|$\)' /etc/pptpd.conf
   option /etc/ppp/options.pptpd
   logwtmp
   localip 10.0.0.88
   remoteip 10.0.0.89-127
   ```

   10.0.0.88 is the internal address of the server, 10.0.0.89-127 is the range that can be used by the pptp clients.

   Then let's see that referred */etc/ppp/options.pptpd*:

   ```
   $ grep -v '^\(#\|$\)' /etc/ppp/options.pptpd
   name pptpd
   refuse-pap
   refuse-chap
   refuse-mschap
   require-mschap-v2
   require-mppe-128
   proxyarp
   debug
   lock
   nobsdcomp
   novj
   novjccomp
   nologfd
   ```

   After everything works fine, you can remove the "debug" line from the config.

   Then add at least one user:

   ```
   # cat /etc/ppp/chap-secrets
   ## client        server  secret                  IP addresses
   mylogin           *       stupidpassword          *
   ```

   The rest is about to allow pptp on the firewall (I'm assuming that you use the default Frugalware configuration: INPUT is on DROP by default, but FORWARD is allowed, OUTPUT too.)

   Add the following 2 lines to the filter section of */etc/sysconfig/firewall*:

   ```
   -A INPUT -p gre -j ACCEPT
   -A INPUT -p tcp -m tcp --dport 1723 -j ACCEPT
   ```

   If you want to allow a client to access Internet via this pptp server, add the following line to the nat section of the same file (change ethX to the correct network interface):

   ```
   -A POSTROUTING -o ethX -j MASQUERADE
   ```

Then check if you have PPP support in the kernel enabled:

```
# lsmod | grep ppp_generic
```

If there is no output, enable it:

```
# modprobe ppp_generic
# echo "ppp_generic" >> /etc/sysconfig/modules
```

Now we're ready to start:

```
# pptpd -f -o /etc/ppp/options.pptpd
```

If no error messages are reported, omit the -f option so it will go background.

Later you can put this to your */etc/rc.d/rc.local*. Debug messages will appear in */var/log/messages* if you're interested in them.

3. Client side

   Install the necessary "pptp" package:

   ```
   # pacman-g2 -S pptp
   ```

   Most howto suggets the pptpconfig (http://pptpclient.sourceforge.net/) tool, it's written in PHP and uses GTK+2. You don't want to use graphical tools locally (and install XOrg) for administrating your machine, do you?

   We can do it by hand, not too complicated.

   You can name every tunnel you create, I'll use here the "mytunnel" name.

   Fire up your favorite editor and create the /etc/ppp/peers/mytunnel file with the following contents:

   ```
   $ grep -v '^\(#\|$\)' /etc/ppp/peers/mytunnel
   name mylogin
   remotename PPTP
   file /etc/ppp/options.pptp
   pty "pptp IP_OF_THE_SERVER --nolaunchpppd "
   require-mppe
   ```

   Your /etc/ppp/chap-secrets should contain the following line:

   ```
   mylogin  PPTP    secret  *
   ```

   We're ready to start the client:

   ```
   # pppd pty 'pptp server --nolaunchpppd' call mytunnel debug dump logfd 2 nodetach
   ```

   A lot of debug messages will be printed, check on an other console if you got a new pppx interface or not:

   ```
   # ifconfig ppp0
   ppp0    Link encap:Point-to-Point Protocol
           inet addr:10.0.0.89  P-t-P:10.0.0.88  Mask:255.255.255.255
           UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:996  Metric:1
           RX packets:7 errors:0 dropped:0 overruns:0 frame:0
           TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:3
           RX bytes:70 (70.0 b)  TX bytes:76 (76.0 b)
   ```

   If it seems to be ok, you no longer need the debug messages and pppd can go backround:

   ```
   # pppd pty 'pptp server --nolaunchpppd' call mytunnel
   ```

   That was all. Not so simple but anyone can do it :-)

4. Resources

   - http://czeh.hu/linuxdoc/vpn-pptp.html - VPN connection using PPTP and Linux by Istvan Czeh (Hungarian)
   - http://webb.gotdns.com:2080/kernel-mppe/pptp-command.html - *pptp-command* HOWTO

17.72 prosody

Don't forget to change /etc/prosody.cfg.lua when needed For more informations about prosody's configurtion, please take a look at : http://prosody.im/doc

If you want to add or delete JIDs you have to be in the *prosody* group You can do it with this command : *usermod -aG prosody LOGIN_NAME*

To start the daemon, type *service prosody start* To automaticly start the daemon at boot time, type *service prosody add* Please do NOT use prosodyctl start and stop

With version 0.9.x, manual changes must be done in the config file, if you migrate from 0.8.x

Please refer to this link for more informations : https://prosody.im/doc/release/0.9.0#upgrading

17.73 psx

Note: You must find a PSX bios on your own, and place it in ~/.pSX/bios.

17.74 pulseaudio

Because PulseAudio can be used as drop-in replacement for ESD you can fool GNOME into loading the PulseAudio daemon just like the traditional ESD daemon. To achieve this use the esdcompat script shipped with PulseAudio. Install pulseaudio-esd : `pacman-g2 -S pulseaudio-esd` Create a symlink from /usr/bin/esd to /usr/bin/esdcompat For more information on `pulseaudio`, please refer to http://www.pulseaudio.org/wiki/PerfectSetup

17.75 pyro

You'll find pyro's scripts in */usr/lib/python2.5/site-packages/Pyro/bin*

17.76 qemu

17.76.1 QuickStart

If you are completely new to `qemu`, you may find the big list of switches a bit confusing. Most users want to install an operating system from a cdrom image to a virtual harddisk. Here is what you need:

```
$ qemu-img create foo.img 8G
$ wget http://server.com/bar.iso
$ qemu -hda foo.img -cdrom bar.iso
```

17.76.2 Guest-agent

The guest agent service is started automatically, as long as the `qemu-guest` subpackage is installed. See here for setup instructions.

17.76.3 Tricks

It worth to read the full documentation at `/usr/share/doc/qemu-*/qemu-doc.html`, it really worth to do so.

To demonstrate how powerful `qemu` is, here are a few cheap tricks:

If you want to be able to ssh to the machine, you can use port redirection. For example using the `-redir tcp:1022::22` option, `qemu:22` will be available at `localhost:1022`.

Note
This requires `root` privileges.

You can create a unix socket to control your virtual machine. For example if you are not able to ssh to the machine, you can still properly shut it down:

Use the `-monitor unix:/tmp/qemu,server,nowait` option, then send the `sendkey ctrl-alt-delete` string to the socket, for example using python:

```
python -c "import socket; sock = socket.socket(socket.AF_UNIX, socket.SOCK_STREAM); \
sock.connect('/tmp/qemu'); \
sock.send('sendkey ctrl-alt-delete\n')"
```

Finally a trick about vnc: using for example the `-vnc 0` option, it's possible to reach qemu's display via vnc. This is quite handy if you run qemu on a server (for example in screen), then you can freely attach to and detach from it whenever you want to do so.

Really, read the full documentation! :)

17.77 quota-tools

To really activate quotas, you'll need to add `usrquota` to the appropriate partitions as listed in `/etc/fstab`. Here's an example:

```
/dev/hda2   /home   ext2   defaults,usrquota    1   1
```

When you want quota support for a given partition, some special files have to be created boot-time. This is not done by default. To do so, you need to

```
# touch /var/lib/quota/new
```

then, reboot to create those files.

To edit user quotas, use `edquota`. See `man edquota`.

17.78 redmine

Post Installation :

Create an empty database and accompanying user named redmine for example.

For Mysql: create database redmine character set utf8; create user *redmine@localhost* identified by *my_password*; grant all privileges on redmine.* to *redmine@localhost*;

For PostegreSQL: create database redmine character set utf8; create user *redmine@localhost* identified by *my_password*; grant all privileges on redmine.* to *redmine@localhost*;

Edit config/database.yml

Generate a session store secret: cd /var/www/html/redmine/ rake config/initializers/session_store.rb

Create the database structure, by running the following command under the application root directory: RAILS_ENV=production rake db:migrate It will create tables and an administrator account.

Insert default configuration data in database, by running the following command: RAILS_ENV=production rake redmine:load_default_d

Fix permissions mkdir tmp public/plugin_assets chown -R redmine:redmine files log tmp public/plugin_assets chmod -R 755 files log tmp public/plugin_assets

Test the installation by running WEBrick web server: ruby script/server webrick -e production see the result : http://localhost:3000/-

- login: admin
- password: admin

SMTP Configuration : Copy config/email.yml.example to config/email.yml and edit this file to adjust your SMTP settings.

see http://www.redmine.org/wiki/redmine/RedmineInstall

For use Apache : http://www.redmine.org/wiki/redmine/HowTo_configure_Apache_to_run_Redmine

17.79 rss2email

17.79.1 Configure:

Create a new feed database:

```
$ r2e new you@yourdomain.com
```

Subscribe to some feeds:

```
$ r2e add http://www.aaronsw.com/2002/rss2email/updates.rss
```

(That's the feed to be notified when there's a new version of rss2email.) Repeat this for each feed you want to subscribe to.

When you run rss2email, it emails you about every story it hasn't seen before. But the first time you run it, that will be every story. To avoid this, you can ask rss2email not to send you any stories the first time you run it:

```
$ r2e run --no-send
```

Then later, you can ask it to email you new stories:

```
$ r2e run
```

You probably want to set this up as a cron job or something.

17.79.2 Customize:

There are a few options, described at the top of rss2email.py. If you want to change something, add it to config.py. For example, to be notified every time a post changes, instead of just once per post:

```
$ echo "TRUST_GUID = 0" >>~/.rss2email/config.py
```

And you can ask rss2email to make the emails look as if they were sent when the item was actually posted:

```
$ echo "DATE_HEADER = 1" >>~/.rss2email/config.py
```

17.80 sawfish

I have included a simple script called sawfish-session which sources $HOME/.sawfish/startup, if it exists. It is setup so you can easily run your own commands before sawfish is launched. You can find an example file at /usr/share/sawfish/startup. Also, there is a sawfish-aplay script as well, which is a wrapper to aplay with the -q argument so your logs aren't spammed by a bunch of useless messages if you choose to use sound events. To use sound events in sawfish, run sawfish-ui and goto the Sound tab, and enable sounds. Then, close the program, run it again, and there should a greyed out box at the bottom for entering a command to for playing sounds. I have disabled ESD support in favor of this. Check the box to enable it, and enter either sawfish-aplay or another program of your choice. However, keep in mind this box cannot accept arguments, it can only accept the path to an executable of some sort, which is the whole reason I included an aplay wrapper. Also, be sure to visit http://sawfish.wikia.com if you want to find stuff to supplement sawfish, like scripts, themes, etc. And, finally, you will an example piece of lisp code you can put in your $HOME/.sawfishrc and edit to your heart's content to get the right root menu for you. This is also where you put

lisp code that you want to become active every time you restart sawfish. Use sawfish-client if you want to test it, and remember to put it in your rc file if you wish to retain it. Happy hacking!

(setq root-menu '(("Editors" ("Abiword" (system "abiword &")) ("Leafpad" (system "leafpad &"))) ("Terminals" ("Sakura" (system "sakura &")) ("xterm" (system "xterm &"))) ("Multimedia" ("Audacious" (system "audacious &")) ("VLC" (system "vlc &"))) ("Network" ("Firefox" (system "firefox &")) ("Pidgin" (system "pidgin &"))) ("Restart" restart) ("Quit" quit)))

17.81 screen

17.81.1 Keeping your screen running across reboots

You may want to restart your screen session automatically after a reboot. This is the case, for example, when we seed the Frugalware ISOs using a torrent client. Here is what you need:

- Set up your `~/.screenrc` so that it'll start your application when screen starts up:

```
screen -t seed 0 /bin/sh -c 'cd $HOME/frugalware-torrents; rtorrent'
```

- Run `crontab -e` and append the following line to your crontab:

```
@reboot screen -d -m
```

You're ready!

17.82 squirrelmail

Please start the *configure* script in the /var/www/squirrelmail directory!

17.83 squirrelmail-check_quota

You have to install this plugin with squirrelmail's own `./configure` tool.

17.84 squirrelmail-login_notes

You have to install this plugin with squirrelmail's own `./configure` tool.

17.85 stunnel

You need some additional configuration before stunnel will be functional:

Adjust the configuration file:

```
# cp /etc/stunnel/stunnel.conf-sample /etc/stunnel/stunnel.conf
# vi /etc/stunnel/stunnel.conf
```

Note
If something goes wrong, try setting `sslVersion` to `all`.

Generate your certificate:

```
# openssl req -new -x509 -days 365 -nodes -config /etc/stunnel/stunnel.cnf -out \
/etc/stunnel/mail.pem -keyout /etc/stunnel/mail.pem
```

Hide the certificate from users:

```
# chmod 600 /etc/stunnel/mail.pem
```

Now you can enable and start the service:

```
# systemctl enable stunnel.service
# systemctl start stunnel.service
```

17.86 sugarcrm

In order to use the sugarcrm, you have to symlink it to somewhere. For example, if you want to use it under http://localhost/-sugarcrm, then use:

```
# ln -s /var/www/SugarSuite /var/www/html/sugarcrm
```

After installing this package, please run in a browser http://localhost/sugarcrm/install.php to setup SugarSuite (sugarcrm).

17.87 syslinux

All the configurable defaults in SYSLINUX can be changed by putting a file called syslinux.cfg.

SYSLINUX searches for the SYSLINUX.CFG file in the following order:

/boot/syslinux/syslinux.cfg /syslinux/syslinux.cfg /syslinux.cfg

Here is a simple example syslinux.cfg file, with one entry to boot a Linux kernel:

DEFAULT linux LABEL linux SAY Now booting the kernel from SYSLINUX... KERNEL vmlinuz.img APPEND ro root=/dev/sda1

see http://syslinux.zytor.com/wiki/index.php/SYSLINUX for the complete documentation.

17.88 trac

After installing trac you need a few steps to set it up. First of all do not forget to install postgresql/mysql/sqlite according to which database backend you want to use.

To create a new trac project, just use the command:

```
$ trac-admin /path/to/myproject initenv
```

You can check the result with:

```
tracd --port 8000 /path/to/myproject
```

Then, fire up a browser and visit http://localhost:8000

For further documentation on trac, how to set up with different HTTP daemons see TracGuide

17.89 tremfusion

Follow this as user:

1) Copy the Tremulous pk3s (data-1.1.0.pk3, vms-1.1.0.pk3, map-atcs-1.1.0.pk3, etc) from their installation directory to /home/<user>/.tr

(Use slocate data-1.1.0.pk3 to find it)

```
$ cp /usr/share/tremulous/base/*.pk3 ~/.tremulous/base/
```

2) Copy z-tremfusion-menu-0.99r3.pk3 to /home/<user>/.tremulous/tremfusion/

(Create the directory if it doesn't exist)

```
$ mkdir ~/.tremulous/tremfusion
$ cp /usr/share/tremulous/tremfusion/*tremfusion*.pk3 ~/.tremulous/tremfusion/
```

3) Copy gamex86.so to /home/<user>/.tremulous/base/

```
$ cp /usr/share/tremulous/base/gamex86.so ~/.tremulous/base/gamex86.so
```

17.90 uget

If you want to use aria2-plugin, first install aria2 package: pacman-g2 -S aria2 == util-linux

17.90.1 Using `tmpfs` for `/tmp`

Frugalware does not use `tmpfs` for `/tmp` by default. However on servers this can cause problems: if you do not reboot for months, then cleaning `/tmp` can take some time. Using `tmpfs` can solve your problem: it's a ramdisk so its content not preserved during a reboot. All you need is to add the following line to your `/etc/fstab`:

```
tmpfs            /tmp            tmpfs       defaults         0   0
```

Note
You need `util-linux >=2.12-31` for this, otherwise X may not start.

17.91 vavoom

17.91.1 Before you play

To be able to play, you must have the IWAD files of the original games and copy it in ~/.vavoom or in /usr/share/vavoom. You can find this IWAD file on the original game CD or in the net. You can use shareware game's IWAD, too.

17.92 vim

If you want to enable spell check support, you need to:

- install the spell files for your language:

```
# pacman-g2 -S vim-spell-xx
```

where `xx` is code of the requested language.

- enable the spell check support for your language (type in `vim`):

```
:setlocal spell spelllang=xx_yy
```

Some languages need correctly set encoding. If you get a message like:

```
Warning: Cannot find word list "hu.latin1.spl" or "hu.ascii.spl"
```

then you need to set your encoding as well:

```
:set encoding=latin2
```

The incorrect words are coloured red by default. You can reach a list of suggested words by pressing `z=` when the cursor is at the given word.

If you want to disable the spell check support, type:

```
:setlocal nospell
```

It may be handy to have map function keys in `~/.vimrc` to enable / disable the spell check support:

```
set encoding=latin2
map <F5> <Esc>:setlocal spell spelllang=en_gb<CR>
map <F6> <Esc>:setlocal spell spelllang=hu<CR>
map <F7> <Esc>:setlocal nospell<CR>
```

Note
The language code is sometimes in an `xx` and sometimes is in an `xx_yy` form. This is something you need to figure out for your language.

See the upstream documentation for more info about spell check support:

```
:help spell
```

17.93 wifi-radar

Don't forget to change the wifi interface name in */etc/wifi-radar.conf*!

17.94 x11vnc

Running `x11vnc` without a password is not recommended. To create one, type:

```
vncpasswd ~/.vnc/passwd
```

Then you can start the VNC server using

```
x11vnc -display :0 -rfbauth ~/.vnc/passwd -forever
```

if are logged in on `:0`.

17.95 xcache

17.95.1 Installing As PHP Extension?

1. Check */etc/php.ini*

```
# cat /usr/share/doc/xcache-$pkgver/xcache.ini >> /etc/php.ini
```

2. Modify php.ini for your needs:

```
# $EDITOR /etc/php.ini
```

3. Restart php

Warning
Use >> with cat, not simply >

Please take a look on xcache wiki.

18 Mailing List Rules

18.1 Introduction

The purpose of this document is to define rules that help the communication on the mailing lists of Frugalware Linux.

18.2 Mailing Lists

THERE ARE 3 READ-ONLY LISTS

- frugalware-announce for general announcements (low traffic)
- frugalware-security for Frugalware Security Advisories
- frugalware-bugs for newly opened tasks in the Bug Tracking System (This may be extended in future, currently you must use the web interface to comment a task.)

THERE ARE 3 LISTS FOR DEVELOPERS

- frugalware-devel for general development questions. Every developer is supposed to read this list. It has a moderate traffic. (Usually only a few mails / day.)
- frugalware-git for Git commits. This is a high traffic list. Every developer is supposed to subscribe to this list, but feel free to set *Mail delivery* to *Disabled* if you don't want to receive mails. (This is required as only subscribed users can post to prevent spam.)
- frugalware-darcs for Darcs patches. No longer used, but we keep this list as the archive is useful sometimes.

THERE ARE 3 LISTS FOR USERS

- frugalware-forums is a bidirectional gateway between the users of the Frugalware Forums (this forum) and people who read the mailing lists only. The primary benefit is that not all developers read the Forums, but mailing lists.

- frugalware-users is for general user questions. It seems the Forums are very popular, but we still provide a mailing list for user questions.
- frugalware-users-hu is for Hungarian user questions.

If not mentioned, then the language of the lists are English. Please use the appropriate language. If you know of other non-English mailing lists, please tell us, then we can include them here.

You can subscribe to our mailing lists here. Also you can unsubscribe or edit your options there.

18.3 Frugalware developers

Developers are supposed to read the `-devel` and `-users` mailing lists, and supposed to be subscribed to the `-git` list.

18.4 Off-list discussion

We don't set a `Reply-to:` header on our mailing lists. This is intentional. If you don't understand why this is a good decision, first please read this document.

In practice if this is a new situation for you, then use your mail client's *list-reply* function, as the *reply* function will send the mail off-list which is not something you want in most cases.

Also please do not use the *group-reply* function if possible. Users must subscribe before they post, so you can be sure they are in the mailing list.

(This is different to some other projects' rules. Some projects require you to use *group-reply* all the time, please do not do so on our lists.)

18.5 Top posting and HTML messages

Please do not top post on our lists. Also please try to avoid HTML messages, many developers use a console mail client to read mails and reading such messages is always problematic.

18.6 Archives

We have our own archive of our mailing lists here. Gmane also provides searchable archives.

19 IRC Rules

19.1 Introduction

This document describes the rules to be followed by everyone who joins the users' and/or developers' IRC channels of Frugalware Linux.

19.2 Welcome

You have joined us on IRC, to get help from or to give help to other Frugalware users. We're sure you have made a good decision :) This document details a few basic rules that should be followed on IRC. The rules are documented here so that they're available to everyone.

19.3 IRC channels

THERE ARE 5 FRUGALWARE LINUX CHANNELS FOR USERS

- #frugalware (Main, English-language only)
- #frugalware.es (Spanish-language only)
- #frugalware.fr (French-language only)
- #frugalware.hu (Hungarian-language only)
- #frugalware.it (Italian-language only)

Please use only the language appropriate to the channel. If you don't do so, you'll be asked to change channels. If you know of other non-English channels, please tell us.

THERE IS A FRUGALWARE LINUX CHANNEL FOR DEVELOPERS

- #frugalware.dev (Frugalware development discussion. Only Frugalware developers can *speak* on this channel but everyone can see what's being discussed).

19.4 Frugalware developers

If you're a Frugalware developer, please also join one or more of the user channels. Since users don't have the right to speak on the #frugalware.dev channel, your presence on a user channel is the only way they can chat with you. Keep in mind that today's Frugalware users may be tomorrow's Frugalware developers.

19.5 Off-topic discussion

19.5.1 Other Linux distributions' features

You may discuss other distributions' features but don't expect everyone to be familiar with them. For example the following question is impossible to answer for someone who hasn't used Gentoo:

```
How can i set up my network so that it works as it does under Gentoo?
```

Instead, describe what it is that you're trying to achieve, for example:

```
Is it possible to use network profiles so that I can change all my settings with one c
ommand when I get home from my workplace?
```

19.5.2 Non-Frugalware discussion

Talking about non-Frugalware topics (or even non-Linux) is okay, as long as this doesn't prevent others from talking about Frugalware. We are a community, so you're welcome to share your ideas, but don't make it impossible for others to get help.

19.6 Asking questions

19.6.1 I'm new to Frugalware

Welcome! You've either installed or are wanting to install Frugalware and so have some general questions. Before asking them in the IRC channel, please read the about page.

19.6.2 First read the Frugalware documentation

Before asking a question, first read the Frugalware documentation to be sure that the answer is not already there. Those who wrote the documentation have spent quite an amount of time and effort. If your question is answered in the documentation you'll be told to read it and provided a link. So please - read the documentation and don't be lazy.

19.6.3 Go ahead and ask

Don't first ask if you can ask a question, just go ahead and ask. The worst that can happen is that you don't get an answer.

19.7 Paste

If you have a few lines of an error message or something similar to show to others in the channel, don't paste it into the channel. This is because (1) IRC is slow and (2) it breaks the flow of other peoples' conversations. Instead, please use our Pastebin, which is available here.

19.8 Is mxw_ a bot?

Yes, it is. It informs users about new binary packages, manages rights on the channel and so on. If you want a new feature to be implemented then feel free to request it at the Frugalware Bug Tracker System (BTS) which is available here.

19.9 Bouncers, leaving your client online when you're away

That's not a problem, but please keep in mind the following: if you are away then you should be able to read back the lines when you were highlighted. If this is not possible then it's better to quit from the channels, since we think that we're talking to you while we're talking with /dev/null. Also if you're online and you have been highlighted and asked, please try to answer. If you have no time, then a simple

Alex: I don't have time ATM to answer, sorry.

is enough. So that he won't wait for your answer.

19.10 Private messaging

Please do *not* /msg users unless you first asked for permission to do so. This is a support channel: you ask in the channel and whoever has the time/knowledge to answer, he/she will. That the fastest way, believe us.

You should also know that some of us (voroskoi, vmiklos, maybe others too) set up their clients to ignore msgs on freenode, so you talk to /dev/null when you /msg to us.

19.11 Logging

All Frugalware channels are logged and public. The logs are linked from the home page, and the main goal is to allow search engines to index them. If you don't like this then your only choice is to not join ;-)

19.12 Verbose away messages, away nicks

Please avoid them, doing so makes the signal-to-noise ratio higher. See the Away messages suck article for further reasons.

20 Checking if Frugalware tarballs are from a trusted source

20.1 How to verify

- Import our public keyring with the following command:

```
$ gpg --recv-keys 20F55619
```

- Verify the tarball. Here is an example:

```
$ gpg --verify pacman-tools-0.7.2.tar.gz.asc pacman-tools-0.7.2.tar.gz
gpg: Signature made Sun 14 May 2006 02:35:34 AM CEST using DSA key ID 20F55619
gpg: Good signature from "Frugalware Linux Archives Verification Key \
        <frugalware-devel@frugalware.org>"
```

20.2 The meaning of this signature

This signature does not guarantee that the Frugalware Linux Archives master site itself has not been compromised. However, if we suffer an intrusion we will revoke the key and post information on the home page as quickly as possible.

21 Creating new packages

21.1 Introduction

Frugalware consists of thousands of packages. Each file in the distribution belongs to a package and you can easily query to which package a file belongs. For example, if you want to know which package contains */etc/frugalware-release*, you should use:

```
$ pacman-g2 -Qo /etc/frugalware-release
/etc/frugalware-release is owned by frugalware 0.6rc1-1
```

If you browse the FST (Frugalware Source Tree), you can see, that in the source directory there are category and category-extra dirs. The dirs without -extra tag contains the basic packages of the given category and the dependencies of the basic packages. So a package in these directories can not depend on a package in extra directories. The same is true for console/graphical applications: if your application/library is graphical, then use xapps/xlib, if not then use apps/lib. For each task there is a default package. For example postfix is our default MTA, so exim, sendmail, etc must be in some extra dir.

The repo has a source and a binary directory. The frugalware repo's directories are source/ and frugalware-$arch/. The binary packages are in the binary directory of the repo. The sources of packages are a little bit more complex. Each package has a category, and each category and package has its own directory in the source dir.

Let's see an example. You are searching for the cabextract package. The binary package is named *frugalware-<arch>/cabextract-<version>-<release>-<arch>.fpm* and its source is placed in the source/apps/cabextract dir.

In the package's own dir, we store everything required to compile the package. You may say we should store only the patches and so, but in our opinion, it's very annoying when you want to recompile a package and the original server is slow or even unreachable, due to some other reasons. Also it may be illegal that we would provide only binary packages without storing the source (since then it may be possible that we are not able to send the source to you even if you ask us by mail).

Besides, there is a FrugalBuild file in each package's source directory. This is a simple bash shell script, that will be included by makepkg. So in the FrugalBuild script you can use everything that can be used in a shell script.

Note
During the package database generation we source all the FrugalBuilds, so it must be a very short time to do so for each FrugalBuild. Because of this, you should not use something like:

```
sha1sums=(`lynx -dump http://foo.com/bar.sha1`)
```

but you should use:

```
# http://foo.com/bar.sha1
sha1sums=('094e3afb2fe8dfe82f63731cdcd3b999f4856cff')
```

This way gensync will be fast even if reaching foo.com takes a lot of time. Also using the `-u` option an offline build is possible.

Briefly, packaging means collecting the sources, adding additional files (for example init scripts or config files) and writing the FrugalBuild script.

21.2 Recompiling packages

Before creating a new package, first we will recompile an existing package in this howto. It's very simple. In our example we will recompile the mplayer package.

First, you have to download the current FST.

- Getting the FST as root

  This is the most simple, you only have to issue:

  ```
  # repoman upd
  ```

- Getting the FST as a simple user

  If you want to do it as a regular user, create the *~/.repoman.conf* file and edit it, change the fst_root dir in it (by default, it would download the files to /var/fst, and it is not writable as a user, of course).

  The *~/.repoman.conf* file should look like:

  ```
  fst_root=~/git
  ```

  Thought fst_root can point to any directory writeable by the user.

  And finally to get the FST, issue:

  ```
  $ repoman upd
  ```

Before building the chroot environment, you should make sure about that the `fst` user exists on your system. Check your */etc/passwd* file. If not, then please check your */etc/passwd.pacnew* file, that contains the relevant entry, just copy that line to */etc/passwd*.

Now that you have the fst user, continue with

```
$ cd $fst_root/source/xapps/mplayer
$ sudo makepkg [<options>]
```

Note
If you are using stable, you probably want to use the `-t stable` option!

First we enter the directory of mplayer then (like make and Makefile) we run makepkg that will build the package according to the parameters described in FrugalBuild. We once had to use the -R option to build the package in a chroot-ed environment. Since 0.5, building in chroot is the default method, you have to use -H if you want to build on the host system. Chroot requires root privileges. To allow a group (for example the devels group) to use sudo makepkg, start visudo as root, and add the following line:

```
%devels ALL=NOPASSWD:/usr/bin/makepkg
```

The chroot will be placed by default in */var/chroot*. Only one package can be built in a chroot at a time, so maybe you'll want to specify a separate chroot for each user. In order to do this, set the $CHROOTDIR variable in your */etc/makepkg.conf* from:

```
export CHROOTDIR="/var/chroot"
```

to

```
export CHROOTDIR="/var/chroot.`echo $HOME|sed \'s|.\*/\(.*\)$|\1|'"`
```

This way the *one parallel build / one system* limit is increased to *one parallel build / one user*.

(See man makepkg for more info about the benefits of building in a chroot).

21.3 Use variables

You can alter the result of the build process using environment variables without touching the FrugalBuild itself. The `git` package is a good example. Using

```
$ sudo makepkg [<options>] USE_DEVEL=y
```

for that package results in a build of git's development version. Here is what you need if you want so for your package:

```
# set the variable to false by default
USE_DEVEL=${USE_DEVEL:-"n"}

(...)

# these commands will be evaluated only in case USE_DEVEL is set to true
if Fuse $USE_DEVEL; then
        _F_scm_type="git"
        _F_scm_url="git://git.kernel.org/pub/scm/git/git.git"
        Finclude scm
fi
```

In the next section we will see an example for a simple FrugalBuild script.

21.4 A simple example

Let's see a simple example, the FrugalBuild script of the cabextract package.

```
# Compiling Time: 0.06 SBU
# Maintainer: James Buren <ryuo@frugalware.org>

pkgname=cabextract
pkgver=1.2
pkgrel=1
pkgdesc="a program to extract Microsoft Cabinet files"
url="http://www.kyz.uklinux.net/cabextract.php"
depends=('glibc')
groups=('apps')
archs=('i686' 'x86_64')
up2date="lynx -dump http://www.kyz.uklinux.net/cabextract.php |grep 'cabextract \
        source code'|tr -s ' '|cut -d ' ' -f 6"
source=(http://www.kyz.uklinux.net/downloads/$pkgname-$pkgver.tar.gz)
```

```
sha1sums=('871b3db4bc2629eb5726659c147aecea1af6a6d0')

# optimization OK
```

And here comes the description for each line:

`#Compiling Time:0.06 SBU`

You should write here how much time it took to build the package. Of course, it depends on your hardware, so we use SBUs instead of minutes as a unit.

SBU is the Static Binutils Unit, which means the time repoman merge binutils takes on your machine. By default makepkg will print out how many seconds the build took. After you built binutils, you should update your */etc/makepkg.conf*:

`SBU="257"`

The line above means compiling binutils on your machine took 257 seconds. From this point, makepkg will print out SBUs instead of seconds after successful builds, and this SBU value will be equal on anyone's machine.

`#Maintainer:James Buren <ryuo@frugalware.org>`

If you are the maintainer of the package, write your name or nick and e-mail address here. If you probably you won't maintain the package, write Contributor instead of Maintainer, and then the Maintainer will add his/her line later. A package may have only one contributor: the first person who wrote FrugalBuild for it. The maintainer is the current maintainer. The other names should not be included in the FrugalBuild, anyone can use the version control features to look for them.

`pkgname=cabextract`

This will be the name of the package. It's allowed to include numbers, hyphens (-), etc., and should be lowercase.

`pkgver=1.2`

The package's version. Hyphens are not allowed, so a 1.0-6111 will be usually converted to 1.0_6111.

`pkgrel=1`

Release number marks Frugalware-specific changes. If you recompile a package, you should increase this number. If you upgrade to a newer version, don't forget to reset this number back to 1. If you design a new package, set this to 1.

`pkgdesc="a program to extract Microsoft Cabinet files"`

A short one-line description for the package. Usually taken from the project's homepage or manpage.

`url="http://www.kyz.uklinux.net/cabextract.php"`

The website of the project.

`depends=(\'glibc\')`

List of dependencies of the package, defined in a bash array. Usually you should compile a package at least two times: first with depends=(), then you should run chkdep -p foo.fpm that will suggest the dependencies, but handle that information with caution! Reading the README, INSTALL and configure.ac files is also a good idea to find out dependencies.

`groups=(\'apps\')`

It is needed to know where, in which category the package belongs. The most important thing: don't put your package in apps, base, devel, lib, multimedia or network, if it depends on X (or on a pkg depending on X, of course). Packages in the extra repository get the *-extra* suffix to the group name.

You should use groups for creating metapackages. The method is the following: put each package to an existing group (group without a hyphen or with the -extra suffix), then add the packages to a new group, something like foo-suite or whatever your want, provided that the name is not an *existing group*.

Example:

`groups=(\'lib-extra\' \'foo-suite\')`

`archs=(\'i686\' \'x86_64\')`

This array defines for which architectures the given package is available. If it's not available, it means that gensync will skip it when generating package databases. If you are not able to provide a binary package for a given arch, don't include that in archs()! For example, no matter if the package could be compiled in x86_64, if you haven't compiled it yourself, don't include it.

```
up2date="lynx -dump http://www.kyz.uklinux.net/cabextract.php |grep 'cabextract \
        source code' |sed 's/.*-\(.*\).t.*/\1/'"
```

A short command that will give us the latest stable version of the package. This helps maintainers to keep the FST up to date. Usually this string consists of three parts: a lynx -dump someurl, a grep foo, and a sed command. We use the http protocol if possible, but sometimes we have to use ftp. In that case instead of `lynx -dump` you should use `wget -O --q`. Of course, you could use wget all the time, but lynx is simpler. The sed command could be replaced with the combination of tr and cut if you prefer them instead of sed. The example used above would be the following with cut and tr:

```
up2date="lynx -dump http://www.kyz.uklinux.net/cabextract.php |grep \
        'cabextractsource code'|tr -s ' '|cut -d ' ' -f 6"
```

`source=(http://www.kyz.uklinux.net/downloads/$pkgname-$pkgver.tar.gz)`

Here you define the sources of the package in a bash array. You can use simple filenames for patches, or additional files when you place them in the same directory as the FrugalBuild script. You can use URLs if you want makepkg to download them automatically. It's important to place all sources in the package's directory including the source files that you can download from a site. Also when dowloading from sourceforge, please use Finclude sourceforge! If you use various random patches from unknown sources, don't expect that somebody else will port those patches to a newer version. You will have to do the work yourself. You have been warned! Actually try to avoid patches unless they are really necessary (eg: secfix, bugfix).

A few words about the size of the sources. If you use an URL then the size is almost unlimited, but if the source is not an url then the source will be added to the FST when the package is accepted. We don't allow files bigger than 100KB in FST. To solve this problem, the sources for a given package are placed in the /pub/other/sources/pkgname dir for each package. If the source is not compressed, we use gzip or bzip2 to compress it first. After this you can use a `http://ftp.frugalware.org/pub/other/sources/pkgname/foo`-styled URL for those big sources.

`sha1sums=(\'094e3afb2fe8dfe82f63731cdcd3b999f4856cff\')`

Another bash array to prevent compiling from the wrong source. Of course this is useless if you just run sha1sum foo.tar.gz after download. Try fetching original sha1sums from the projects website, if possible. It's a good idea to leave a comment above this line about where to find these sha1sums.

As you can see there in no build() function in this FB. It's because we wrote some F* functions to make our work easier. It's something similar you can see in Gentoo for example. These functions can be found in *source/include/util.sh* file inside the FST. An empty build actually means:

```
build() {
        Fpatchall
        Fmake "$@"
        Fmakeinstall
        if echo ${source[@]}|grep -q README.Frugalware; then
                Fdoc README.Frugalware
        fi
}
```

So Fpatchall will apply all the patches in source() array, then Fmake calls the configure script and make command, then Fmakeinstall acts like make install, finally if a README.Frugalware file is given it will also add that to the package. For details see the utils.sh file, it's well documented.

Note

You don't have to use these F* commands, but we **highly** recommend it. Also if you use simple commands do not forget to add || return 1 after each command, so the build will stop on error!

`#optimization OK`

This line will be added automatically to the end of the FrugalBuild if the build() function used your $CFLAGS or $CXXFLAGS. This is handy if you want to cross-compile on a faster machine for a slower architecture. If the package doesn't use our $CFLAGS we can't cross-compile it, so please try to avoid creating "unoptimized" packages. If the package doesn't contain any architecture-dependent file, then you can add this line manually as makepkg will not detect this.

21.5 Full reference

Now here is a full list of directives available.

First, let's start with the install directive. Here you can refer to an install file (usually $pkgname.install) to use. If there is a $pkgname.install in the FrugalBuild's directory, it will be used automatically. In the install file, you can define actions to be executed before/after installing/upgrading/removing the package. A skeleton of this file can be found under /docs/skel in FST.

Of course, you probably will not need all of these functions, just remove what you don't need. If you want to do exactly the same after upgrading as after installing, feel free to use post_install $1 in the post_upgrade() function.

Save this file as $pkgname.install, thus makepkg will automatically use it. You should not specify the install script in the source array as it is not used in build().

The pkgname, pkgver, pkgrel, url, source and sha1sums directives were discussed in the previous section.

The backup array is used to make some files in the package as config files. If possible, we don't modify config files during an upgrade. Example:

```
backup=(\'etc/pacman-g2.conf\')
```

Note that the leading slash is missing!

For more information about this, see the handling config files section in the pacman-g2 manpage

The depends array has been discussed already, except I haven't mentioned before that the elements may include version information, for example:

```
pkgname=kdewebdev
depends=('kdelibs=3.3.0')
```

Here you can use <>, ⇐, >= or = operators.

The makedepends array defines packages required only in build time. For example if the source is in SRPM format, probably alien is a build-time requirement.

The rodepends array defines packages required only in runtime. It must be used in any case when putting the given package in the depends() array would cause circular dependency.

In the conflicts array, you can define a list of packages that shouldn't be installed if you want to install this package. Let's see an another example:

```
pkgname=mutt-devel
conflicts=('mutt')
```

It is necessary as the two packages are almost the same, but the binaries differ. In this case the mutt package must also contain this line: `conflicts=(\'mutt-devel\')`. Of course, if two or more packages conflict eachother, only one of them can be placed in a non-extra group.

The provides array is used to create virtual dependencies. It means both postfix and sendmail provides mta, so we can do:

```
pkgname=mailman
rodepends=('mta')
```

The user has a choice between postfix and sendmail.

The last one in this list is the replaces directive. The module-init-tools package is a good example:

```
pkgname=module-init-tools
replaces=('modutils')
conflicts=('modutils')
```

As you can see, we often make such new packages which also conflict with each other. Using the replaces directive when users use pacman-g2 -Su next time, if modutils is installed (probably :)), they will be asked to remove modutils and install module-init-tools.

```
license=(\'GPL2\')
```

This directive is optional. At the moment, you may add such a field, but copy the LICENSE field from the source root to the packages's documentation dir, so this isn't really necessary.

21.6 Subpackages

Since 0.5 makepkg can also create subpackages. It is very useful when your package has graphical parts based on qt for example. It's a pain for gnome users as they want the package, but they do not want the qt part. So you create a subpackage for qt part and both side is happy. Let's see an example:

```
# Compiling Time: 1.43 SBU
# Maintainer: crazy <crazy@frugalware.org>

pkgname=djvulibre
pkgver=3.5.18
pkgrel=2
pkgdesc="DjVu is a web-centric format for distributing documents and images."
depends=('libtiff' 'libjpeg')
makedepends=('kdelibs' 'gnome-mime-data' 'gnome-icon-theme' 'htop')
rodepends=('xdg-utils')
groups=('xapps')
archs=('i686' 'x86_64')
options=('scriptlet')
_F_sourceforge_dirname="djvu"
_F_sourceforge_broken_up2date=1
Finclude sourceforge
url="http://djvulibre.djvuzone.org/"
source=(${source[@]} head_-n1.patch  no-OPTS-FLAGS-thx.patch)

subpkgs=('djview')
subdescs=('DjVu viewer for qt and mozilla plugins.')
subdepends=('libxi libgl qt libxmu')
subrodepends=('djvulibre')
subgroups=('xapps-extra')
subarchs=('i686 x86_64')

build()
{
        Fcd
        Fpatchall
        Fautoreconf
        export CFLAGS="$CFLAGS"
        export CXXFLAGS="$CXXFLAGS"
        Fconf \
                --enable-threads \
                --disable-desktopfiles \
                --enable-xmltools \
                --enable-djview
        make depend || Fdie
        make || Fdie
        Fmakeinstall
        Fln /usr/lib/netscape/plugins/nsdejavu.so \
                /usr/lib/mozilla/plugins/nsdejavu.so
        Fln djview3.1.gz usr/share/man/man1/djview.1

        Fsplit djview usr/bin/djview
        Fsplit djview usr/bin/djview3
        Fsplit djview usr/lib/mozilla
        Fsplit djview usr/lib/netscape
        for i in . ja; do
                [[ $i == . ]] && Fsplit djview usr/share/man/$i/man1/djview.1
                Fsplit djview usr/share/man/$i/man1/djview3.1
                Fsplit djview usr/share/man/$i/man1/nsdejavu.1
        done
        Fsplit djview usr/share/djvu/djview3
}
```

Here you can see the djvulibre FrugalBuild. Note subpkgs, subdescs, subdepends, subgroups and subarchs. These 5 value is lethal for a subpackage. There are other subpackage variables too of course. See `man FrugalBuild` for details. Also note that bash does not support two-dimensional arrays, so when defining the array of arrays, then quotes are the major separators and spaces are the minor ones.

Defining the subpackage is only the first part of creating a subpackage. You have to tell makepkg which files you want to put in the subpackage. We use Fsplit command for this. First parameter is the subpackage name, second is the file you want to move. Please never use a trailing slash when defining file patterns, especially if you use wildcards in it!

If you need more example just take a look on avahi FrugalBuild in network group.

Note

Use subpackages when they are necessary, but do not start making foo-devel, foo-common, foo-not-so-common, foo-quite-common-but-not-that-common packages :) Making too much subpackage makes maintaining too hard and simplicity is the frugal way.

21.7 Compiling the package

That's fairly simple. In the package directory you should do exactly the same as described in the Recompiling packages section. If you want to contribute this package to the Frugalware project, then go to BTS, open a feature request and upload each non-downloadable file (ie. FrugalBuild, install scriptlet, patches) as an attachement. Please do not forget to check your FrugalBuild with *fblint* command before uploading it. Fblint is available in pacman-tools package.

Happy packaging!

21.8 Kernel modules

A few words about kernel modules. They're special as even if you installed the correct version of the kernel (and kernel-source) package, sometimes the modules are compiled for the running kernel, so you have to check if compiling against other kernel version than the running one works or not. You can use the modinfo command for this. If crosscompiling does not work always add Fcheckkernel to the build(). So here is the list of conditions a kernel module package have to satisfy:

1) Should depend on kernel=version, where version is the version of the kernel defined in *$fst_root/source/include/kernel-module.sh*. (Always use up-to-date FST!)

2) Should Finclude the kernel-module scheme.

3) If you want to use a custom install script (saying running just depmod -a after the install/upgrade is not enough for you) then the install script should run depmod -a. Otherwise the scheme will provide so a scriptlet which does so.

4) build() should call *Fcheckkernel* to ensure the module will be compiled for the right kernel version or it should be commented if you have checked the compiling for other kernel version. It is good for out build servers as they may not run the kernel provided by the given package tree. (They can't run -stable *and* -current kernels at the same time :)).

5) Kernel modules may be installed for the not-currently-running kernel. To ensure they are registered properly, you need to use the Fbuild_kernelmod_scriptlet function. It generates the proper install scriptlet for you.

See `man kernel-module.sh` for more info.

21.9 Repoman

Repoman is simple tool to download all packages' buildscript and compile programs from source.

The most commonly used repoman commands are the following:

```
repoman merge package
```

or simply

```
repoman m package
```

builds a package from source and installs it. You can configure the build options in the makepkg_opts directive of */etc/repoman.conf*.

By default repoman will install the missing dependencies with pacman, clean up the leftover work files, install the package, and write the resulting package to the current working directory.

```
repoman update
```

or simply

```
repoman upd
```

updates FST in /var/fst (or the directory set in *~/.repoman.conf*). First time repoman will download it (it may take some time!).

22 This is a small tutorial for those who want to contribute to Frugalware

22.1 Ways of contributing

There are many different ways to contribute to Frugalware. You can write documentation, translate the existing documentation into your native language (or any other language you want to), maintain packages or improve them with added features etc.

If you are a programmer you can help us in developing our applications. These are: pacman-g2, gfpm, fwlive, frugalwareutils, setup etc. See git.frugalware.org for different project repositories.

You can also start new projects. If you show some code we can surely host your project too if it's Frugalware related. For example you want to write kfpm :)

Important
After each title in brackets you can find the target audience.

22.1.1 Translations (translators)

You can read the details on our Translations documentation page.

22.1.2 Necessary documentation (packagers, coders)

In the first part I will cover the information necessary for those who do *not* have developer status yet.

In the second part we will set up the necessary config files.

First of all, we ask you to read the following documentation carefully. If you do not want to deal with packages, but just want to code it's usually enough to read the git documentation as we store our code in git repositories.

- man makepkg
- man pacman-g2
- man repoman
- man FrugalBuild
- man fwmakepkg
- makepkg howto
- git getting started

I know, it is boring reading documentation, but you have to know that writing it is even worse so do not ask questions when the answer in the documentation. If you can not understand something feel free to join #frugalware@irc.freenode.net and ask.

22.1.3 Downloading and setting up the repositories

Getting the frugalware-current repo (packagers)

The frugalware-current repo is the development repo for the packages.

When you want to get it you need the git package. Let's get it:

```
# pacman-g2 -S git
```

Now create a git directory where you can hold all your repos. You can choose any other name of course.

```
$ mkdir -p ~/git
$ cd ~/git
```

Now clone the repo with git:

```
$ git clone http://frugalware.org/git/pub/frugalware/frugalware-current current
$ cd current
```

Now be patient while git clones all the objects and then checks out the files. Also you can use other mirrors as well.

Getting pacman-g2 and other code (coders)

First of all you need the repo of the program. In this example I will use pacman-g2, but the steps are very similar. NOTE: Most of our programs need the translations repo to compile)

```
$ mkdir -p ~/git
$ cd ~/git
$ git clone http://frugalware.org/git/pub/other/translations (optional)
$ git clone http://frugalware.org/git/pub/other/pacman-g2/pacman-g2
$ cd pacman-g2
```

Setting up the repository and sending patch via email (packagers, coders)

Now you should setup up your identity.

```
$ git config --global user.name "Your Name"
$ git config --global user.email email@addr.ess
$ git config branch.master.rebase true
```

Now you can make your changes. When finished run

```
$ git diff
```

in the repository.

Tip
You can also use *git diff .* (note the dot in the end). In that case git will show the changes recursively in the current directory. It is very handy when you have lot of uncommitted changes in your repo.

If you are satisfied with the changes run

```
$ git commit -a
```

to commit all your changes.

If you want to cherry-pick hunks from your changes:

```
$ dg record
```

or using native git commands:

```
$ git add -p; git commit
```

Without committing your changes you can not send nor push (just developers) it.

Tip
With frugalware-* repos it's recommended to use *repoman rec* which is a wrapper for dg record. It sets up the patch name properly so you only need to deal with the details.

Note
Here you can find more details on how to write good commit messages in general using git.

Here comes the final step. Send in the patch(es)!

```
$ git format-patch <hash>
$ git send-email --to frugalware-devel@frugalware.org .
```

`<hash>` is the `sha1` of the last patch you do *not* want to submit. Run

```
$ git log
```

and you'll see the hash. Also, you can just use your existing mail client and send the patch(es) as an attachment.

If everything goes fine your patch should show up on the frugalware-devel mailing list soon.

Note
You have to subscribe to the frugalware-devel mailing list and set up your SMTP server properly (if you use `git send-email`).

It doesn't really belong to here but I want to document it somewhere. If you are a developer and want to apply such a patch, you need:

- Check the patch itself. If the second line is not an empty one, then you need to hand-edit the patch before applying:

```
Subject: [PATCH] powwow-1.2.13-1-i686
 * new package
```

to:

```
Subject: [PATCH] powwow-1.2.13-1-i686

* new package
```

- Then you can apply the patch using git-am:

```
$ cat 0002-powwow-1.2.13-1-i686.patch | git am
```

You should do this in the root directory of the repository.

22.1.4 Further options for those who have developer account (packagers, coders)

Once you get a developer account, you have the right to request the following services:

- BTS access (so that we can assign tasks to you)
- git write access (you'll always get this, except if you are working on the artwork or so)
- a @frugalware.org mail address (with imaps/pop3s access)
- Public and private devspace. The first is in the `/pub/other/people/nick` dir and this is mirrored (you must not put private stuff to there). The later is your `~/public_html` dir: it is not mirrored and there is no backup for it. Though you may temporarily put private stuff to there.
- a @frugalware.org jabber account if you want one

What you should do:

- You should read the frugalware-devel mailing list. When you're asked, please try to respond.
- If you push patches to git, you should subscribe to the frugalware-git mailing list. This is list has a big traffic since a new mail is sent for each patch. If you don't have time to read it, subscribe then set the "I would like to receive no mail" option. Also take care that your subscribing email address is the same one you set using `git config user.email`
- It's good if you can join the user and developer channel when you're online.
- Maintain your packages. Try to resolve your assigned bugs, try to keep your packages up to date, and if you needed patches for packages, send them upstream. If you don't have anything to do for a week that's usually a bad sign. It's - of course - OK when you go for vacation a few times a year, but then please announce it on the developer mailing list so that we won't wait for you when fixing urgent problems, etc.
- Document your work. The documentation is worth nothing if it's outdated. Ideally somone who has never contacted us should be able to understand every detail of Frugalware, just from documentation. No secrets! We are not kids.
- If you have time, try to read the mailing lists (`frugalware-users*@`) and the forums. If you prefer reading the forums from your mail client, there is a bi-directional gateway on the `frugalware-forums@` list, use it.

Let us see what you should set up to get it work. I will also give some tips which can make your life easier.

Read this page, we collected a set of tricks when we converted from darcs to git.

Setting up the frugalware-* repos and repoman (packagers)

It is time to set up some necessary things. We start with the frugalware-current repo. Make sure that you are in the root of the frugalware-current repo. Also do not forget to change the username to your login name on git.frugalware.org.

```
$ git config remote.origin.url 'username@git.frugalware.org:/home/ftp/pub/frugalware/ ↩
    frugalware-current'
$ git config remote.origin.receivepack "sudo -u repo git-receive-pack"
```

As you will use `repoman` to upload the packages (and many other things as you'll see) we should set it up now. This step is also necessary. Open *~/.repoman.conf* with your favourite editor and add the following lines:

```
fst_root=~/git
current_servers=("username@git.frugalware.org:/home/ftp/pub/frugalware/frugalware-current")
stable_servers=("username@git.frugalware.org:/home/ftp/pub/frugalware/frugalware-stable")
stable_pushonly="y"
```

Where fst_root is the directory where you store your git repos. Username is your login on git.frugalware.org. For details see `man repoman`.

As from now use the following command from package's directory to push your changes.

```
$ repoman push
```

It will check the FrugalBuild using fblint, then record your changes, push them, upload the fpms and finally create the changelog, update the fdb etc. So you are done if there was no error message.

Setting up other repos (coders)

In repo's main directory:

```
$ git config remote.origin.url 'username@git.frugalware.org:/home/ftp/pub/other/pacman-g2/ ↩
    pacman-g2'
$ git config remote.origin.receivepack "sudo -u owner git-receive-pack"
```

Do not forget to change the username and repository path. For paths refer to the gitweb interface.

Note
The owner for `pacman-g2`, `frugalwareutils`, `pacman-tools` is usually `vmiklos`.

You should always review what you would push before you perform the action:

```
$ git fetch
$ git rebase origin/master
$ git log origin/master..master
```

Then you can use

```
$ git push
```

to send in your changes.

Note
The `dg push` wrapper does exactly this for you.

---

23 Security support

23.1 Introduction

This document documents the work of the Frugalware Security Team. Primarily it's for new developers or for existing developers who join the Security Team.

23.2 Handling security bugs

1. The security team opens a new task in the BTS, with a [SEC] prefix.
2. The maintainer fixes the issue in -current and decides if the issue needs fixing in -stable or not. If yes, then changes the status of the task to "Fixed in -current", otherwise closes the task.
3. If there is no patch for the issue yet, then set the status to "Researching". This indicates that you, the maintainer, is aware of the problem, but don't yet have enough a solution.
4. The security team regularly searches for "Fixed in -current" bugs, fixes the issue in -stable and releases a new FSA.

23.3 How to release an FSA?

1. Check if the backport built by syncpkgd is ready (the binary packages should be uploaded for each arch).
2. Open the -stable Changelog file of the package. There you can see the vulnerable and unaffected versions of the package.
3. Add a new entry to the frugalware/xml/security.xml file in the homepage-ng repo.
4. Commit, push. The commit hook will check if the xml is valid, so most common errors can be avoided. In rare cases, the announcement may not appear on the frugalware-security list. If this is the case, then ask on -devel about what the problem might be.
5. Close the task in the BTS, filing in FSAxxx in the closure message.

23.4 How to notice security issues

1. Subscribe to Secunia Security Advisories List at http://secunia.com/secunia_security_advisories/ page. This is the best place to notice issues.
2. Read the mails one-by-one and check if the advisory affects -current or -stable.
3. Open a task in BTS if necessary. Please fill in the form correctly, provide a patch if you can.

You can also read other mailing lists, like https://lists.grok.org.uk/mailman/listinfo/full-disclosure, but Secunia monitors them, so you won't miss anything. (You just notice things later.)

23.5 How to get patches

Secunia announces security issues days after they released so there is a good chance to find a patch.

1. First of all sometimes upstream fixes it with a new version.
2. Fixed in cvs/svn/whatever and you are able to find the patch (unlike PHP)
3. If these two fail, there is http://security.ubuntu.com/ubuntu/pool. Secunia also mails you if the bug fixen in Ubuntu, so steal the patch from them :) You only need the $package-$pkgver.diff.gz. There is a changelog in it, where you can find the filename of the fix.
4. It's also a good idea to take a look on RedHat/Gentoo bugzilla. They attach fixes most the time.

So it's good to read the Secunia mails carefully as you'll always know when the patch is available.

23.6 Versioning

We use integers in pkgrels for normal packages, but -stable updates are different. Here are the cases:

- If you do a version bump (we refer to them as *secfix bump* usually in -stable commit messages), then you need to set pkgrel to 1<release_codename>1.
- If you add a security patch, and pkgrel was an integer (let's say *1*), then you should increment pkgrel to 2<release_codename>1.
- If you add a security patch when the pkgrel was already in an X<release_codename>Y form, increment it to X<release_codename>Y+1 (Alternatively, you can use X+1<release_codename>Y if there is already a newer version in -current.)

This ensures that:

- The version of the security update will be larger than the one in -stable, so that the package will be upgraded when the user does a pacman-g2 -Syu on -stable.
- The version of the security update will be smaller than the one in -current, so that the package will be upgraded when the user upgrades to a new version (current or new stable).

24 Handling git repositories

24.1 Introduction

This document is for developers who want to publish a git repository on the Frugalware FTP Server and on the Frugalware Gitweb Interface.

24.2 Name of the repository

The name of the official repositories are `frugalware-current`, `frugalware-stable` and so on.

The name of WIP repositories are typically in a `featureNUM` form, like `kde45` or `parted2`, referring to the name of the software it contains and its version. This method is used so that the repository name can be a valid shell variable as well.

Please note that there is a convention that WIP repository names never contain a hyphen (-). This is on purpose. It's not trivial to decide that when you merge code from one repository to another then build servers should try to build automatically the new packages you brought in or not. Because of this the policy is that if a hyphen is in the name, the it'll build the new packages (WIP → -current merge), but it won't do so when you merge the other way around.

24.3 Location of the repository

Since a repository consists of plain files, we can and should place them on the ftp server (/home/ftp). To prevent further problems, always use the server name "git.frugalware.org", currently it's an alias of genesis.frugalware.org.

First decide if it's a personal repository or a team one. For example if you create a repository to update to a newer python version, then you will probably do all the work, create it under `/pub/other/people/nick/reponame`. Simply create a dir, issue `git init` and push at least one commit to there (but before pushing, enable the hooks, see below).

Now anyone can `git clone` it, using a *full mirror*, for example `ftp://ftp12.frugalware.org/mirrors/ftp.frugalware.org/pub/`.

24.4 Registering for the gitweb interface

If the repository is a team one, then create it under /pub/other. In this case you probably want the gitweb interface, too. To use it:

1. Update the file `.git/description` inside the repo with a short (less than 80 chars) description.
2. Create the file `.git/owner` inside the repo containing your name, *without* your email address.
3. Push a *relative* symlink to the `homepage-ng` repository, see the existing ones as a reference.

After some time (a maximum of 30 minutes) it should appear at `http://git.frugalware.org/`.

24.5 Enabling hooks for your repository

Currently you need hooks for the following reasons: . If you don't use *bare* repositories, then the content outside `.git` won't be updated automatically, you need a hook to do so.

1. If you want CIA notification.
2. If you want to send mails to the `Frugalware-git` mailing list.
3. If you want to let others clone your repository via *dumb* protocols like `http` or `rsync`. (This means that if you disable this hook, it won't be accessible anonymously!)

For the last one:

```
mv .git/hooks/post-update{.sample,}
echo "unset GIT_DIR; cd ..; git checkout -f" > .git/hooks/post-receive
chmod +x .git/hooks/post-receive
```

For the others:

```
ln -sf /home/ftp/pub/other/git-hooks/git-hooks.py .git/hooks/post-receive
```

One thing that a hook won't do for you is to allow pushing to the master branch, even if it's the checked out one. This is normally not good, but our hook will handle this, so we can ignore the problem:

```
git config receive.denyCurrentBranch ignore
```

24.6 Setting up server configuration for a WIP repo

When you run repoman, it invokes `repoman server` on the remote machine using ssh. `repoman server`, just like plain repoman, reads configuration from `/etc/repoman.conf` and `$HOME/.repoman.conf`, so you need to set up the later before you can push packages to your WIP repo.

Here is a minimal example:

```
fst_root=/home/nick/git
repos=('current' 'mywiprepo')
```

And then you have to symlink the repos to `$HOME/git`, for example:

```
cd $HOME/git
ln -s /pub/frugalware/frugalware-current current
ln -s /pub/other/people/nick/nicktesting/ nicktesting
```

24.7 Enabling syncpkgd support for a WIP repo

If you create a new WIP repo, syncpkgd won't sync packages in it by default.

This means that if you just push your commits, no attempt will be made to build the relevant binary package automatically for you, which is the case for the -current / -stable repos.

If you want syncpkgd support, then you need to edit 3 configuration files on the server which runs `syncpkgd` (that's typically not your local machine and not the one that runs `syncpkgcd`).

Edit syncpkgd's repoman config by extending the repos array and adding the foo_servers, foo_sudo and foo_bases variables:

```
vi ~syncpkgd/.repoman.conf
```

Add a pacman-g2 configuration file:

```
vi ~syncpkgd/.pacman-g2/repos/foo
```

The contents will be something like this:

```
[foo]
Server = http://ftp.frugalware.org/pub/other/people/nick/foo/frugalware-@CARCH@
```

Note
Don't replace `@CARCH@` with anything else, syncpkgcd will do so later!

Finally edit the git hook and add foo to the end of the repos array:

```
vi /pub/other/git-hooks/synchook/config.py
```

If you no longer need these entries, you can remove them, but leave at least one there as an example.

25 GNOME Bump HOWTO

You **MUST** follow this HOWTO when bumping GNOME to a new version (even a minor version).

To start, packages must be compiled in the order listed below (if you find a change that needs to be made to this list, poke Bouleetbil). If it is a major bump (2.14 to 2.16, for example), it is wise to rebuild most of the GNOME packages.

25.1 GNOME compile order

- libxml2
- libxslt
- gnome-common
- intltool
- rarian
- gtk-doc
- glib
- libIDL
- ORBit2
- libbonobo
- fontconfig
- Render
- Xrender
- cairo
- cairomm
- Xft
- pango
- atk
- shared-mime-info
- gtk*
- gtk+2-engines
- gtkmm
- gconf
- desktop-file-utils
- gnome-mime-data
- avahi
- avahi-glib
- dbus
- hal

- gamin
- dbus-glib
- libgnome-keyring
- gnome-keyring
- libproxy
- libsoup
- gvfs
- gnome-vfs
- audiofile
- esd
- libgnome
- libart_lgpl
- libglade
- libgnomecanvas
- libbonoboui
- hicolor-icon-theme
- icon-naming-utils
- gnome-icon-theme
- libgnomeui
- startup-notification
- gnome-themes
- gnome-doc-utils
- gnome-desktop
- libwnck
- libgpg-error
- libgcrypt
- libtasn1
- opencdk
- gnutls
- firefox
- libgweather
- evolution-data-server
- pygobject (*)
- pycairo
- pygtk (*)

- gnome-menus
- librsvg
- libcanberra-gtk
- gnome-panel
- zenity
- metacity
- gstreamer
- liboil
- libxklavier
- libgnomekbd
- libcroco
- eel
- gst-plugins-base
- gnome-settings-daemon
- nautilus
- control-center
- gnome-session
- vte
- gnome-terminal
- libgtop
- gucharmap
- gnome-applets
- libgsf
- libgnomecups
- libgnomeprint
- libgnomeprintui
- yelp
- bug-buddy
- gtksourceview
- pygtksourceview
- pyorbit (*)
- gnome-python (*)
- iso-codes
- totem-pl-parser
- totem

- brasero
- gnome-media
- eog
- poppler
- evince
- gedit
- gnome-python-desktop
- alacarte
- nautilus-cd-burner
- gst-plugins-good
- libmusicbrainz
- gconf-editor
- gnome-utils
- gnome-system-monitor
- gnome-netstatus
- gcalctool
- at-spi
- libgail-gnome
- gnome-speech
- gnome-mag
- gnopernicus (missing from repo)
- gok (missing from repo)
- epiphany
- epiphany-extensions
- gob2
- gnome-games
- gnome-user-docs
- file-roller
- gnome-nettool
- vino
- vinagre
- gnome-volume-manager
- gnome-backgrounds
- sound-juicer
- gtkhtml

- gal
- pilot-link (if needed, not a gnome part)
- gnome-pilot
- gnome-pilot-conduits
- gnome-spell
- evolution
- evolution-webcal
- evolution-exchange
- gdm
- ptlib
- opal
- ekiga
- dasher
- gnome-power-manager
- gnome-keyring-manager
- deskbar-applet
- fast-user-switch-applet
- gnome-screensaver
- pessulus
- sabayon
- gnome-cups-manager
- system-tools-backends
- liboobs
- cheese
- gnome-system-tools
- mousetweaks
- seahorse
- gnome-sharp
- gnome-desktop-sharp
- empathy
- hamster-applet
- nautilus-sendto

(*) - don't use Fsplit on this package.

Note
all *sharp and all bindings need to be split

25.2 Bumping individual packages

Never, I repeat, **NEVER** bump a version without doing the following:

1. Download the new version's tarball and extract it
2. Run ./configure --help and look in configure.in to check for new dependencies (even optional ones) and consider whether to use them or not. Consult all devels about whether it is a good idea to use the optional dependencies.
3. Check for dependencies that are no longer needed and remove them from the FrugalBuild
4. Check GConf schemas. Sometimes they have been renamed, or new ones have been added. Not doing this can cause a lot of problems.
5. Check the Changelog and NEWS file for the package. Sometimes there may be API/ABI changes that need to be considered before bumping.
6. Check if `\_F_gnome_{scrollkeeper,mime,desktop}` are needed in the new version.
7. When all this has been done, update the FrugalBuild with new sha1sums, pkgver, depends, GConf schemas and `\_F_gnome_*` values (add gnome-scriptlet to Finclude if necessary)
8. Build the package and push.

26 KDE Bump HOWTO

26.1 KDE4

You **MUST** follow this HOWTO when bumping KDE4 to a new version (even a minor version).

First, update KDE4 version (_F_kde_ver) in source/include/kde-version.sh. Commit this (git commit kde.sh). From kde-4.7.0 we store all the sha1sums in source/include/kde-version.sh so don't forget to update that as well (you can use this to fetch the sha1sums).

Next, packages must be compiled in the order listed below (if you find a change that needs to be made to this list, poke the kde m8r). You can generally find the sha1sums in the kde website (somewhere like http://kde.org/info/KdeVersion.php). If it is a major bump (i.e. 4.6 to 4.7) it is wise to check the updated build instructions on the Kde website, and work in a kde testing repo first.

KDE4 compile order

- kdelibs (1)
- kdepimlibs (1)
  - kfilemetadata
  - baloo
  - baloo-widgets
  - nepomuk-core
  - nepomuk-widgets
  - kactivities
- kdebase (1)
- kdebase-runtime (1)
- kde-base-artwork
- kdebase-workspace (1)

- konsole
- kdebase-workspace-wallpapers (-extra)
- svgpart (-extra)
- kdenetwork (virtual package)
  - kdenetwork-filesharing
  - kdenetwork-strigi-analyzers
  - zeroconf-ioslave
  - kget
  - kopete
  - kppp
  - krdc
  - krfb
- kdegraphics (-extra) (virtual package)
  - kruler
  - kolourpaint
  - libkipi
  - libkexiv2
  - libkdcraw
  - libksane
  - mobipocket
  - gwenview
  - kcolorchooser
  - kamera
  - okular
  - ksaneplugin
  - kdegraphics-thumbnailers
  - kdegraphics-strigi-analyzer
  - kgamma
  - ksnapshot
- libkcddb
- libkcompactdisc
- kdemultimedia
- libkomparediff2 (-extra)
- kdesdk (-extra)
- kdewebdev (-extra)
  - analitza (-extra)
  - libkdeedu (-extra)
  - kqtquickcharts (-extra)
- kdeedu (-extra)

  - pykde4 (-extra)
- kate (-extra)
- kdebindings (-extra) (virtual package) (2)
  - smokegen
  - smokeqt
  - smokekde
  - perlqt
  - perlkde
  - qtruby
  - korundum
  - kross-interpreters
  - qyoto
  - kimono
- kdeutils
- kdeadmin (-extra)
- kdeplasma-addons (-extra)
- kdeaccessibility (-extra)
- kdeartwork (-extra)
- kdetoys (-extra)
  - amor
  - kteatime
  - ktux
- libkdegames (-extra)
- libkmahjongg (-extra)
  - kdegames (-extra)
- kdepim-runtime
- kdepim (-extra)
- kde-l10n (3)
- oxygen-icons (4)

(1) = these 5 packages **MUST** be built first, and in **THIS** order.

(-extra) = kde-extra packages. Some are needed as makedepends for other packages.

(2) = packages split from kdebinding must be built with **THIS** order.

(3) kde-l10n is usually built last.

(4) oxygen-icons can be updated at anytime.

26.2 KDE5

KDE5 is splitted into Frameworks (KF5), Plasma and applications. Up to now only KF5 has been packaged, plasma is next.

26.2.1 KF5

As for kde4, update frameworks version in source/include/kde-version.sh (_F_kdever_frameworks). Sha1sums are stored in the same file so don't forget to update them as well. Packages must be compiled in the order listed below (if you find a change that needs to be made to this list, poke the kde m8r). You can generally find the sha1sums in the kde website (somewhere like http://kde.org/info/kde-frameworks-KF5Version.php).

KF5 compile order

- extra-cmake-modules
- attica-qt5
- kitemmodels
- kitemviews
- kplotting
- threadweaver
- kcodecs
- kguiaddons
- kidletime
- kwidgetsaddons
- sonnet
- kconfig
- kwindowsystem
- solid
- kglobalaccel
- karchive
- kdbusaddons
- kcoreaddons
- kjs
- kimageformats
- kauth
- kcrash
- kjobwidgets
- kcompletion
- ki18n
- kdoctools
- kdnssd
- kconfigwidgets
- kservice
- kiconthemes

- knotifications
- kwallet5
- kpty
- kemoticons
- kdesu
- ktextwidgets
- kxmlgui
- kbookmarks
- kcmutils
- kio
- kparts
- kdewebkit
- kdesignerplugin
- knewstuff
- kdeclarative
- kinit
- kded
- knotifyconfig
- kunitconversion
- kjsembed
- kross
- kmediaplayer
- ktexteditor
- kapidox
- frameworkintegration
- kdelibs4support
- khtml
- kactivities5
- plasma-framework
- krunner5

27 Frugalware Release HOWTO

27.1 Introduction

The aim of this howto is to show what's the procedure of a stable Frugalware release. The to-be-created release in this howto is 2.0, the previous release is 1.9.

27.2 A testing release

A testing release is similar to a full one, but much simpler. Here are the steps:

- bump the `frugalware` package: update the Makefile in `frugalware.git`, upload a new release tarball, and update the package in -current
- rebuild the `setup` package, update the version of the `frugalware` package dependency to the new version
- wait for the nightly cronjob to publish setup kernel+initrd under `/pub/frugalware/frugalware-current/boot`
- now you can generate a netinstall iso using mkiso for a single architecture you can test and upload the image to `/pub/frugalware/frugalware-current-iso`
- do a default install and make sure the machine boots up and you can log in using the graphical interface (if not, then fix it)
- run `dg tag <version>` for the new version and push, using:

```
git push
git push --tags
```

- sync changes from -current to -testing:

```
$ rsync -avP --delete-after frugalware-current/ frugalware-testing/
```

- generate installer images for a single architecture using `mkisorelease`
- wait at least 24h so that mirrors will be in sync
- update news.xml and roadmap.xml to mark the release as done

27.3 Preparing

- send a mail to -devel about "please stop version and release bumps"
- check if the artwork has been updated completely. see this mail from Nadfoka on what items should be checked
- ask someone to update the screenshots
- sync the archs, checkpkgs shouldn't have any red pkg in it's output
- run gensync to rebuild the fdbs
- generate isos and test if everything is ok (ie. install from cd1-cd2 on i686, and start kde, or something)
- check if the upgrade from 1.9→2.0 works or not, probably a simple -Syu is not enough, then write a howto
- tag the release using git tag

27.4 Creating the stable tree

Copy the full tree on genesis:

```
$ cd /home/ftp/pub/frugalware
$ cp -av frugalware-current frugalware-2.0
```

27.5 Updating the -current tree

Now one has two trees. All what one should do in -current is to regenerate ChangeLog.txt (copy & paste the command from `tools/genpkgdbs`).

27.6 Updating the -stable tree

- run tools/mkpkglst for each arch
- update VERSION in docs/Makefile, and rebuild the manual
- update `\.git/description`
- run genpkgs to regenerate the ChangeLog.txt to start from the 1.9 tag to the 2.0 tag
- rename the frugalware-current fdbs to rigel:

```
for i in frugalware-*; do cd $i; mv frugalware-current.fdb rigel.fdb; cd ..; done
```

- tweak the syncpkgd config, so that the rigel repo will be recognized as a non-current-based WIP repo
- search for *ugly* in genesis:/pub/other/git-hooks/synchook/synchook.py and make it reply 2.0 with rigel as well.
- you must place a copy of the new repo file in each build server's `/etc/pacman-g2/repos` directory or else it will not work.
- update pacman-{g2,-tools} and fwsetup so that -stable will be the default on -Syu / repoman upd / in the installer, not -current — and git push these changes (see previous stable release if you need an example!)
- once all archs are ready with the "default to -stable" builds, rename rigel fdbs to frugalware:

```
for i in frugalware-*; do cd $i; mv rigel.fdb frugalware.fdb; cd ..; done
```

- upload the fdbs to the mysql db using fpm2db, just run all2db.sh from the /tools dir
- update the frugalware-stable and frugalware-stable-iso symlinks

```
ln -sfn frugalware-2.0     frugalware-stable
ln -sfn frugalware-2.0-iso frugalware-stable-iso
```

- create a new chroot tarball for each arch

27.7 Testing

- generate isos, test *all* of them (net,cd,dvd for each arch)
- create an usb stick installer tarball for each arch
- create an tftp boot image for each arch
- create a gui installer image for each arch

27.8 Announcement

- put the isos online and wait at least 24h so that the mirrors will be in sync at release time
- create torrents for the isos and make sure at least one machine seeds them
- add the new version to the bts
- write an announcement, put it out to somewhere and ask Alex or LGee to spellcheck it
- push it to the homepage-ng repo
- mark the release as "done" in `/frugalware/xml/roadmap.xml` (homepage-ng repo) and add the proper newsid value
- update the topic of `#frugalware`
- update the freecode.com entry

27.9 For the next release

- find a codename
- update roadmap.xml

Done!

28 Artwork requirements

28.1 Introduction

This document details the requirements that must be met by all artwork if it is to be accepted into the official Frugalware gallery.

28.2 The rules

- All artwork must be licensed under the Free Art License 1.3 (full details).
- Where the Frugalware logo appears, only the officially approved logo may be used. Refer here for the logo.

Note
There is a newer SVG version available here.

- Artwork must be submitted in either SVG or XCF (The Gimp) format as this allows for derivative works to be made without affecting the impact of the original artwork. Examples of derivative works include wallpapers in various sizes and height/width ratios, and/or KDM/GDM/SLiM themes. To suit the varying sizes and ratios of monitors, any wallpaper must be a minimum 1600 pixels wide and provided in both 4:3 and 16:9 ratios.
- All artwork must be submitted together with any associated source files - i.e. files which are required by the graphics editor used by the entrant to reproduce and/or edit the artwork.
- Only FLOSS software may be used to create the wallpaper.
- Neither the release's version number, nor code-name are to appear in artwork, or there should be a version without them for later use when a given release is no longer supported.

29 Table of user / group ids used in Frugalware

Table 1: Users and groups that are added with a specific uid/gid

ID	User	Package	Group	Package
000	root	shadow	root	shadow
001	bin	shadow	bin	shadow
002	daemon	shadow	daemon	shadow
003	adm	shadow	sys	shadow
004	lp	shadow	adm	shadow
005	sync	shadow	tty	shadow
006	shutdown	shadow	disk	shadow
007	halt	shadow	lp	shadow
008	mail	shadow	mem	shadow
009	news	shadow	kmem	shadow
010	uucp	shadow	wheel	shadow
011	operator	shadow	floppy	shadow
012	syncpkgd	pacman-tools	mail	shadow
013			news	shadow
014	ftp	shadow	uucp	shadow
015			man	shadow
016			cdrom	shadow
017			scanner	shadow
018	privoxy	privoxy	privoxy	privoxy
019	fst	pacman	audio	shadow
020	nx	freenx	games	shadow
021			slocate	slocate
022			utmp	shadow
023			camera	shadow
024			video	shadow
025	smmsp	shadow	smmsp	shadow
026	clamav	clamav	clamav	clamav
027	mysql	shadow	mysql	shadow
028	rsyncd	rsync	rsyncd	rsync
029	_ntp	openntpd	_ntp	openntpd
030			storage	shadow
031	pgdb	postgresql	pgdb	postgresql
032	rpc	shadow	rpc	shadow
033	sshd	shadow	sshd	shadow
034	scponly	scponly	scponly	scponly
035			sbox	scratchbox
036			rlocate	rlocate
037			netdev	shadow
038	messagebus	dbus	messagebus	dbus
039	hald	hal	hald	hal
040	amavis	amavisd-new	amavis	amavisd-new
041	ejabberd	ejabberd	ejabberd	ejabberd
042	gdm	shadow	gdm	shadow
043			shadow	shadow
044	beagleindex	beagle	beagleindex	beagle
045	partimag	partimage	partimag	partimage
046	sabayon	sabayon	sabayon	sabayon
047	munin	munin and munin-node	munin	munin and munin-node
048			ccache	ccache
049	openldap	openldap	openldap	openldap
050			ftp	shadow
051			telnetd	shadow
052			tape	shadow

Table 1: (continued)

ID	User	Package	Group	Package
053			dialout	shadow
054	prosody	prosody	prosody	prosody
055			lock	systemd
056	gitosis	gitosis	gitosis	gitosis
057			systemd-journal	systemd
058	systemd-jounal-gateway	systemd	systemd-journal-gateway	systemd
059				
060			grsec_procview	kernel-grsec
061			grsec_audit	kernel-grsec
062			grsec_tpe	kernel-grsec
063			grsec_s_all	kernel-grsec
064			grsec_s_client	kernel-grsec
065			grsec_s_server	kernel-grsec
066	mediatomb	mediatomb	mediatomb	mediatomb
067	polkituser	policykit	polkituser	policykit
068	usbmuxd	usbmuxd	usbmuxd	usbmuxd
069	couchdb	couchdb	couchdb	couchdb
070				
071				
072				
073	postfix	postfix	postfix	postfix
074				
075			postdrop	postfix
076				
077	dspam	dspam	dspam	dspam
078				
079				
080	mailman	mailman	mailman	mailman
081				
082	exim	exim	exim	exim
083				
084	avahi	avahi	avahi	avahi
085	firebird	firebird	firebird	firebird
086				
087				
088				
089				
090	pop	shadow	pop	shadow
091				
092				
093				
094				
095				
096				
097				
098			nobody	shadow
099	nobody	shadow	nogroups	shadow
100			users	shadow
101		shadow	console	shadow
102				
103				
104	distccd	distcc	distccd	distcc
105				

Table 1: (continued)

ID	User	Package	Group	Package
106				
107				
108				
109	postgrey	postgrey		
110				
111				
112				
113	logcheck	logcheck	logcheck	logcheck
114				
115				
116				
117				
118				
119				
120				
121				
122				
123				
124				
125				
126				
127				
128				
129				
130				
131				
132				
133				
134				
135				
136				
137				
138				
139				
140				
141				
142				
143				
144				
145				
146				
147				
148				
149				
150	quagga	quagga	quagga	quagga
151				
152				
153				
154				
155				
156				
157				
158				
159				

Table 1: (continued)

ID	User	Package	Group	Package
160				
161				
162				
163				
164				
165				
166				
167				
168				
169				
170				
171				
172				
173				
174				
175				
176				
177				
178				
179				
180				
181				
182				
183				
184				
185				
186				
187				
188				
189				
190				
191				
192				
193				
194				
195				
196				
197				
198				
199				
200				
201				
202				
203				
204				
205				
206				
207				
208				
209				
210				
211				
212				
213				

Table 1: (continued)

ID	User	Package	Group	Package
214				
215				
216				
217				
218				
219				
220				
221				
222				
223				
224				
225				
226				
227				
228				
229				
230				
231				
232				
233				
234				
235				
236				
237				
238				
239				
240				
241				
242				
243				
244				
245				
246				
247				
248				
249				
250				
251				
252				
253				
254				
255				
256				
257				
258				
259				
260				
261				
262				
263				
264				
265				
266				
267				

Table 1: (continued)

ID	User	Package	Group	Package
268				
269				
270				
271				
272				
273				
274				
275				
276				
277				
278				
279				
280				
281				
282				
283				
284				
285				
286				
287				
288				
289				
290				
291				
292				
293				
294				
295				
296				
297				
298				
299				
300			jupiter	jupiter
301				
302				
303				
304				
305				
306				
307				
308				
309				
310				
311				
312				
313				
314				
315				
316				
317				
318				
319				
320				
321				

Table 1: (continued)

ID	User	Package	Group	Package
322				
323				
324				
325				
326				
327				
328				
329				
330				
331				
332				
333				
334				
335				
336				
337				
338				
339				
340				
341				
342				
343				
344				
345				
346				
347				
348				
349				
350				
351				
352				
353				
354				
355				
356				
357				
358				
359				
360				
361				
362				
363				
364				
365				
366				
367				
368				
369				
370				
371				
372				
373				
374				
375				

Table 1: (continued)

ID	User	Package	Group	Package
376				
377				
378				
379				
380				
381				
382				
383				
384				
385				
386				
387				
388				
389				
390				
391				
392				
393				
394				
395				
396				
397				
398				
399				
400				
401				
402				
403				
404				
405				
406				
407				
408				
409				
410				
411				
412				
413				
414				
415				
416				
417				
418				
419				
420				
421				
422				
423				
424				
425				
426				
427				
428				
429				

Table 1: (continued)

ID	User	Package	Group	Package
430				
431				
432				
433				
434				
435				
436				
437				
438				
439				
440				
441				
442				
443				
444				
445				
446				
447				
448				
449				
450				
451				
452				
453				
454				
455				
456				
457				
458				
459				
460				
461				
462				
463				
464				
465				
466				
467				
468				
469				
470				
471				
472				
473				
474				
475				
476				
477				
478				
479				
480				
481				
482				
483				

Table 1: (continued)

ID	User	Package	Group	Package
484				
485				
486				
487				
488				
489				
490				
491				
492				
493				
494				
495				
496				
497				
498				
499				
500				
501				
502				
503	bitlbee	bitlbee	bitlbee	bitlbee
504				
505				
506				
507				
508				
509				
510				
511				
512				
513				
514				
515				
516				
517				
518				
519				
520				
521				
522				
523				
524				
525				
526				
527				
528				
529				
530				
531				
532				
533				
534				
535				
536				
537				

Table 1: (continued)

ID	User	Package	Group	Package
538				
539				
540				
541				
542				
543				
544				
545				
546				
547				
548				
549				
550				
551				
552				
553				
554				
555				
556				
557				
558				
559				
560				
561				
562				
563				
564				
565				
566				
567				
568				
569				
570				
571				
572				
573				
574				
575				
576				
577				
578				
579				
580				
581				
582				
583				
584				
585				
586				
587				
588				
589				
590				
591				

Table 1: (continued)

ID	User	Package	Group	Package
592				
593				
594				
595				
596				
597				
598				
599				
600				
601				
602				
603				
604				
605				
606				
607				
608				
609				
610				
611				
612				
613				
614				
615				
616				
617				
618				
619				
620				
621				
622				
623				
624				
625				
626				
627				
628				
629				
630				
631				
632				
633				
634				
635				
636				
637				
638				
639				
640				
641				
642				
643				
644				
645				

Table 1: (continued)

ID	User	Package	Group	Package
646				
647				
648				
649				
650				
651				
652				
653				
654				
655				
656				
657				
658				
659				
660				
661				
662				
663				
664				
665				
666				
667				
668				
669				
670				
671				
672				
673				
674				
675				
676				
677				
678				
679				
680				
681				
682				
683				
684				
685				
686				
687				
688				
689				
690				
691				
692				
693				
694				
695				
696				
697				
698				
699				

Table 1: (continued)

ID	User	Package	Group	Package
700				
701				
702				
703				
704				
705				
706				
707				
708				
709				
710				
711				
712				
713				
714				
715				
716				
717				
718				
719				
720				
721				
722				
723				
724				
725				
726				
727				
728				
729				
730				
731				
732				
733				
734				
735				
736				
737				
738				
739				
740				
741				
742				
743				
744				
745				
746				
747				
748				
749				
750				
751				
752				
753				

Table 1: (continued)

ID	User	Package	Group	Package
754				
755				
756				
757				
758				
759				
760				
761				
762				
763				
764				
765				
766				
767				
768				
769				
770				
771				
772				
773				
774				
775				
776				
777				
778				
779				
780				
781				
782				
783				
784				
785				
786				
787				
788				
789				
790				
791				
792				
793				
794				
795				
796				
797				
798				
799				
800				
801				
802				
803				
804				
805				
806				
807				

Table 1: (continued)

ID	User	Package	Group	Package
808				
809				
810				
811				
812				
813				
814				
815				
816				
817				
818				
819				
820				
821				
822				
823				
824				
825				
826				
827				
828				
829				
830				
831				
832				
833				
834				
835				
836				
837				
838				
839				
840				
841				
842				
843				
844				
845				
846				
847				
848				
849				
850				
851				
852				
853				
854				
855				
856				
857				
858				
859				
860				
861				

Table 1: (continued)

ID	User	Package	Group	Package
862				
863				
864				
865				
866				
867				
868				
869				
870				
871				
872				
873				
874				
875				
876				
877				
878				
879				
880				
881				
882				
883				
884				
885				
886				
887				
888				
889				
890				
891				
892				
893				
894				
895				
896				
897				
898				
899				
900				
901				
902				
903				
904				
905				
906				
907				
908				
909				
910				
911				
912				
913				
914				
915				

Table 1: (continued)

ID	User	Package	Group	Package
916				
917				
918				
919				
920				
921				
922				
923				
924				
925				
926				
927				
928				
929				
930				
931				
932				
933				
934				
935				
936				
937				
938				
939				
940				
941				
942				
943				
944				
945				
946				
947				
948				
949				
950				
951				
952				
953				
954				
955				
956				
957				
958				
959				
960				
961				
962				
963				
964				
965				
966				
967				
968				
969				

Table 1: (continued)

ID	User	Package	Group	Package
970				
971				
972				
973				
974				
975				
976				
977				
978				
979				
980				
981				
982				
983				
984				
985				
986				
987				
988				
989				
990				
991				
992				
993				
994				
995				
996				
997				
998				
999				

30 List of packages needs to be rebuilt after the given bumped

Note
In general, if you want to see the list of missing rebuilds, run `./checkabi` from the tools directory (provided that on the machine in question the SQL db is filled with ABI info).

30.1 kernel

For current:

```
revdep-rebuild 276
```

If you want syncpkgd to do the job:

```
revdep-rebuild 276 --nobuild --nopush
```

Note
Please use this only on minor (ie. 2.6.22.1 → 2.6.22.2) bumps, on a major bump many packages need fixing manually.

For solaria:

```
revdep-rebuild 41222 -t stable --nobuild --nopush
```

30.2 mysql

Only in case sover increases, for example if you update to 5.5.10:

```
git grep 'depends.*libmysqlclient>='|grep -v 5.5.10
```

30.3 libgda

(maybe need rebuild)

- gnumeric
- libgnomedb

30.4 db

(only on major bumps, ie. 4.2.x → 4.3.x)

```
$ git grep "'db>="
```

about 28 packages at the moment.

30.5 gnutls

- bitlbee (.so)
- claws-mail
- filezilla
- kildclient
- lftp
- libpurple (pidgin)
- libsoup (NOTE: first libsoup bump then all the other gnome | gtk* apps)
  - bug-buddy
  - evolution-data-server
  - rhythmbox
  - seahorse
  - swfdec
  - vino
- liferea

- msmtp
- net6
- python-gnutls
- weechat
- wireshark (.so)

30.6 dbus

- hal
- evince
- gnome-utils
- gnome-media
- gnome-volume-manager
- nautilus-cd-burner
- ivman
- k3b
- pmount
- kdebase
- xfce4-terminal
- liferea
- bmpx
- bluez-libs

30.7 dbus-mono

- banshee
- tomboy
- f-spot
- galago-sharp

30.8 neon

- subversion
- rpm
- openoffice.org
- gst-plugins-bad
- fusedav

30.9 binutils

- amule

30.10 libtasn1

- gnutls
- evolution (need to figure out which part depends on libtasn1 . . .)
- lftp
- libsoup
- loudmouth

30.11 gstreamer

(only if is an upgrade for example, from 0.8 to 0.10, or 0.10 to 0.12, etc)

- amarok
- banshee
- rhythmbox
- totem
- gnome-applets
- gnome-control-center
- and probably a lot of gnome too

30.12 gtk+2

(only need for special version bumps. Example 2.8 → 2.10 we need bump these packs because /usr/lib/gtk+-2.0/1.X.X directory changed. BTW not at all bumps. Ex.: 2.6→2.8)

- gtk+2-engines
- librsvg
- libgnomeui
- gtk-xfce-engines
- kde-theme-qtcurve

30.13 libcdio

- sound-juicer

30.14 vte

- gnome-terminal
- xfce4-terminal
- gtk2-sharp
- anjuta
- tilda
- grip
- awn-extras-applets
- guake
- mlview
- roxterm
- ruby-gnome2
- gnome-desktop-sharp
- cairo-dock-plugins
- geany
- sakura
- sjterm
- termit
- nemiver
- lxterminal

30.15 firefox

To rebuild packages for a new version, bump the up2date in `source/include/firefox-i18n.sh`, then:

```
cd source/locale-extra/
for i in $(ls -d firefox-*|egrep -v 'spell|dict')
do
        cd $i
        bumppkg && repoman rec "- version bump"
        cd - >/dev/null
done
```

30.16 xulrunner

- galeon
- epiphany
- devhelp
- yelp

30.17 wireless_tools

- kdenetwork

30.18 parted

To rebuild packages for `parted-1.8.8`:

```
revdep-rebuild 429 --other --sed "s|'parted[^']*'|'parted>=1.8.8'|"
```

30.19 libpqxx

- kpogre
- asterisk-addons
- asterisk
- koffice

30.20 openobex

- kdebluetooth

30.21 bluez-libs

- bluez-utils
- kdebluetooth
- libbtctl
- gnome-bluetooth
- bluez-pin

30.22 gail

(.so version bump)

- eel
- gtkhtml

30.23 imagemagick

- dvdauthor

30.24 evolution-data-server

- ekiga
- evolution

30.25 x264

- mplayer
- avidemux
- ffmpeg
  - libquicktime
  - baresip
  - mplayer2
  - vlc

30.26 ocaml

- facile

30.27 openbox

- obconf

30.28 pilot-link

- gnome-pilot
- gnome-pilot-conduits
- libmal
- kdepim
- evolution
- sylpheed-claws

30.29 php

- eaccelerator

30.30 libevent

(on sover change)

- tor
- nfs-utils
- trickle

30.31 exiv2

- gwenview
- libkexiv2
- digikam
- kipi-plugins
- kphotoalbum

30.32 icu4c

- boost
- libtorrent-rasterbar
- texlive
- tin
- webkit

30.33 c-ares

- aria2
- php
- bzflag
- xine-ui
- sword

30.34 libofx

- homebank

30.35 directfb

- gst-plugins-bad
- splashy

30.36 sword

- bibletime

30.37 gpm

- fpc
- joe
- vim
- pycrypto
- jed
- xemacs
- fte
- links
- elinks
- aumix
- aalib

30.38 libtorrent-rasterbar

- qbittorrent
- flush
- springlobby

30.39 ghc

- haskell-bytestring-show
- haskell-mmap
- haskell-primitive
- haskell-agum
- haskell-hunit
- haskell-zlib
- haskell-html
- haskell-dataenc
- haskell-syb
- haskell-tar
- haskell-parallel
- haskell-text
- haskell-utf8-string
- haskell-random
- haskell-mtl
- haskell-dlist

- haskell-extensible-exceptions
- haskell-data-default-class
- haskell-hashed-storage
- haskell-tf-random
- haskell-quickcheck
- haskell-regex-base
- haskell-regex-posix
- haskell-regex-compat
- haskell-parsec
- haskell-network
- haskell-hslogger
- haskell-http
- haskell-vector
- haskell-data-default-instances-dlist
- haskell-data-default-instances-containers
- haskell-data-default-instances-old-locale
- haskell-data-default-instances-base
- haskell-data-default
- haskell-x11
- xmonad
- xmonad-contrib

31 Creating translations for init scripts

Marcus Habermehl <bmh1980de@yahoo.de>

31.1 Preparing the source

To make a script translatable you must first add these two lines to the rc script.

```
TEXTDOMAIN=my_service
TEXTDOMAINDIR=/lib/initscripts/messages
```

To mark a string as translatable in bash you must prefix the string with $.

```
echo $"This is a translatable string."
```

31.2 Creating the pot file

After this you must create the pot file.

```
$ bash --dump-po-strings rc.my_service | xgettext -L PO -o rc.my_service.pot -
```

31.3 Creating a po file

In the next step you create the po file.

```
$ msginit -l hu_HU
```

Now you can edit the po file with any editor.

31.4 Creating the mo files

To create and install the mo files, you must add the po files to the source() array and use the `Frcd2` macro in build().

32 Frugalware Asciidoc quickstart

Since 0.6 Frugalware, all documentation is written in Asciidoc which means we have to write README.Frugalware files in Asciidoc syntax. Here are some basic Asciidoc features and some things you should and should not do a README.Frugalware.

32.1 Features

You can use *bold*, _italic_ and also `monospaced` fonts.

You can also ``quote'' if you want to do so.

When you want to add something to the

```
 ------------
 # root command line
 $ user command line
 > keyboard input
 ------------
```

that's no problem at all.

Maybe you want bulleted items:

```
 .Items
 * item 1
 * item 2
 * here is number 3
```

And you can also create lists:

```
1. First
+
It's indented, belongs to first.
+
And this paragraph is also indented.

2. Second
+
This is inside the second point.
+
2.1. Foo
+
2.2. Bar
+
a. Baz

3. Third

End of list.
```

Some extras:

```
NOTE: You can also place notes.

TIP: It's a tip

WARNING: Warning.

IMPORTANT: This is important

CAUTION: Cave canem!
```

32.2 Restrictions

You **must not** underline titles with = or -. You might use ~, and ^ for subchapters. If you want one line titles place 3 or 4 = before the title and a space.

32.3 Skeleton for README.Frugalwares

Your titles should look similar to this:

```
=== First chapter

--------------
# pacman-g2 -Syu
--------------

=== Second one

'\_F_foobar'

==== This is a subchapter...

...and its contents.
```

or

```
First chapter
~~~~~~~~~~~~~

--------------
# pacman-g2 -Syu
--------------

Second one
~~~~~~~~~~

'\_F_foobar'

This is a subchapter...
^^^^^^^^^^^^^^^^^^^^^^

...and its contents.
```

32.4 Skeleton for standalone documentation

You might ask then: okay, but how do I start? Here is a really simple example:

```
= Title
Author Name <foo@frugalware.org>

== First chapter

--------------
pacman-g2 -Syu
--------------

== Second one

`\_F_foobar`
```

And you can generate the HTML using

```
asciidoc -a toc -a numbered skel.txt
```

The documentation should be placed under the `/docs` dir in the FST. Please add a link to it in `index.txt` and in `index-user.txt` or `index-devel.txt` depending on the type of the documentation.

32.5 Buiding it on your own machine

Install the tools necessary to build the documentation (if you haven't already done so):

```
# pacman-g2 -S make asciidoc po4a
```

Get the necessary source code and translations:

```
$ mkdir ~/git
$ cd ~/git
$ git clone http://frugalware.org/git/pub/frugalware/frugalware-current current
$ git clone http://frugalware.org/git/pub/other/translations
$ cd current/docs
```

Generate additional documentation and update the po files from the translations repository:

```
$ make packages.txt user.txt po
```

Generate the localized documentation source from the po files:

```
$ po4a -k 0 po4a.cfg
```

Generate HTML from the source:

```
$ cd hu
$ asciidoc -a toc -a numbered -a sectids network.txt
```

Now you can look at the result of your translation in a web browser.

If you have already done this, and you updated the translation, you need to:

```
$ cd ~/git/translations
$ git pull --rebase
$ cd ~/git/current/docs
$ rm -rf po
$ make po
$ po4a -k 0 po4a.cfg
$ cd hu
$ asciidoc -a toc -a numbered -a sectids network.txt
```

and now you should be able to see your updated translation in the updated HTML.

32.6 Adding a new project to Pootle

Well, this happens rarely, and so is not well documented, but here is what is needed:

- `autogen.sh` should support importing po files from the `translations` repository and should have a `--pot-only` switch. `gnetconfig` is a good example.
- The pot file should be updated daily. Add the project's `autogen.sh` to -current's `/tools/genpkgdbs`.
- Run the above command manually once.
- Add the pot file to `pootle-update` in the `pacman-tools` repository.
- Run `pootle-update` manually once.
- Log in to Pootle with administrator rights and create a new project.
- Add the necessary new languages on the web interface.
- Translate a few strings for one language and commit.
- Pull the translations repository locally and verify that you get the expected results.

33 Frequently Asked Developer Questions

33.1 What is the recommended way to version bump a package if I don't have git push access?

a. Update the FrugalBuild.

b. Optional: update the patches/docs/etc.

c. Compile the package.

d. Upload the new .fpm to incoming.

e. repoman rec, git format-patch and git send-email the fixes. (Don't forget to set your git identity!)

33.2 makepkg ends up with <packagename>: /usr/info/dir: exists in filesystem

Instead of

```
make DESTDIR=$startdir/pkg install
```

you should write

```
Fmakeinstall
```

in your FrugalBuild.

33.3 I can't pacman-g2 -Su <package>, it says local version is newer, but I know it isn't!

This is a bug in the package's version numbering, so please report this in the Bug Tracker System. Since pacman-g2 checks the version numbers (installed vs. repo version), the new package's version must be bigger than the old one to upgrade flawlessly.

33.4 What does 5.55 SBU mean?

It took 5.55 times longer for the maintainer to compile this package than binutils. So if you want to know how long it will take to compile a package with 5.55 SBU, you should first compile binutils (makepkg helps you, as it writes how many seconds elapsed). Then you should multiply it by 5.55 to know how many seconds it will take to compile the package.

33.5 Why do maintainers cry about my new package's tarball?

Let's have a look at the filelist of eaccelerator's tarball:

```
$ tar -tf eaccelerator-0.9.3-1.tar.bz2
eaccelerator/
eaccelerator/eaccelerator-0.9.3.zip
eaccelerator/FrugalBuild
eaccelerator/README.Frugalware
eaccelerator/eaccelerator-0.9.3-1-i686.fpm
```

You have to name the tarball as <pkgname>-<pkgver>-<pkgrel>.tar.bz2 (or gz), which should only contain a <pkgname> directory at first level, and all the files needed to create the fpm in it. It is the easiest way for the maintainers to work with your tarball when adding your package to the repo.

33.6 What should and shouldn't I include in depends(), rodepends() and makedepends()?

You should include only what chkdep -p recommends, and avoid trivial makedepends, including:

- auto*
- make
- gcc
- kernel-headers
- libtool
- glibc

Don't forget: every depends is a makedepends as well!

The rodepends() array should only contain packages really needed for running the given application.

33.7 What are the various dependancy-control arrays for?

- *depends* should contain any packages that this one depends on at compile and run time as well.
- *makedepends* is for packages that this one needs to compile.
- *rodepends* is for run time only dependencies; eg. a wordlist package (with no executables) needs a program which can handle it as a dictionary.
- *provides* is an alternate name for the package. Main use is for more packages which do the same; eg. hunspell-en and hunspell-de both provide hunspell-dict, and hunspell depends on hunspell-dict instead of any specific language. (Sometimes those packages are conflicting, like postfix provides *and* conflicts with mta, and exim too - this way there can be only one MTA on the system, without the need to know other MTAs' name.)

Be careful with dependency-cycles: while pacman-g2 can handle them, makepkg can not.

33.8 How can I have PHP to work with my newly packaged eaccelerator/anything extension?

Since package A should not tamper with package B's config files,you should write a README.Frugalware, describing how to enable/use the extension, include it in source() and Fdoc README.Frugalware.

33.9 How can I cross-compile (package) an architecture-independent (non-binary) program?

You should modify carch and chost in */etc/makepkg.conf* and build the package again.

33.10 repoman upd can't create /var/fst/ as it already exists

`Su` – to root and

```
cd /var/fst && mv * frugalware-current
```

33.11 How can I access the central FW repo (mirrors are too slow for me)?

```
git clone http://git.frugalware.org/repos/frugalware-current
```

This creates a new local repo for you, which is a copy of the central repo. To update it, run

```
git pull --rebase
```

in it. That's all to have a read-only copy; if you want to git send-email patches, then you should read the Git docs to set up your name, email, etc.

33.12 What should I write as patch name and long comment at repoman rec?

Patch name should be the same as the fpm (but without .fpm, of course); and long comment should only contain what you have done to create that patch (eg. "added i686 to archs()").

33.13 Where should I place my comments about a package?

You mean README.Frugalware. It should be in source() and then at the end of the build() you should use:

Fdoc README.Frugalware

It is automatically included if you use empty build() or Fbuild.

33.14 I want to work with the latest development version of pacman&co.! How?

```
$ git clone http://git.frugalware.org/repos/pacman-tools
$ cd pacman-tools
$ make dist
```

You will have a brand new .tar.gz. Give it to pacman-tools' FrugalBuild, correct the checksum, create a new pacman-tools package (makepkg -fucH helps) and install it. That's all (and if you don't understand this, read it again, and if it's still not clear, then wait for pacman-tools' normal upgrade since you don't need this really)...

33.15 Naming locale packages

What is the order of a new package's locales? How should I name them?

Have a look at hunspell There is a hunspell package, which depends on hunspell-dict. There is no package named hunspell-dict, but it is provided by the locale packages. The most important ones are -en (==en_US), -hu (==hu_HU), -de (==de_DE), -fr (==fr_FR), -it (==it_IT), -es (==es_ES) and -sk (==sk_SK). Here are others: -en_US, -de_CH, -es_MX.

The -xx packages will be installed by the non-CD based (ie. netinst, DVD) installers.

33.16 Error handling

You are responsible for checking if a command used in build() fails. The best is to use the F* macros where possible since they handle the errors for you. If you need custom commands, it's recommended to append `|| return 1` to every line, so that build() will stop if an error occurs.

33.17 Permissions

If text files (header files, documentation) are executable, feel free to fix their permission. A bigger problem is the permission of the shared libraries. They must be executable, please fix their permission if necessary. As always, it's recommended to create a patch to fix the problem and send it to the upstream project.

33.18 Stripping

Stripping binaries is unnecessary and pointless. Unless you use `options=(\'nostrip\')` in the FrugalBuild, it's done by makepkg automatically.

33.19 When should I use $Fsrcdir and $Fdestdir

Most F* macros will prepend/append those variables for you, but if you use custom commands, then you always have to use them.

33.20 When should I increment a package's release number?

- If your change affects only the FrugalBuild (like an up2date fix) then you should not, just push your change.
- If your change affects the fdb or the fpm (change in build(), depends() fix, etc) you should do so.

33.21 How do I repair a corrupted package database?

Restore a backup from the `/pub/other/fdb-snapshot` directory, and check its version (the `.version` file in the tarball).

Then run:

```
$ for i in `git log --pretty=oneline 94a41e0..|sed 's/^[^ ]* \([^ ]*\).*/\1/'\
        |sed 's/-[^-]*-[^-]*-[^-]*$//'`; do ls ../source/*/$i &>/dev/null \
        || continue; updatesync upd frugalware-current.fdb \
        ../source/*/$i/FrugalBuild; done
```

34 Frugalware Source Tree Testsuite

34.1 Introduction

The testsuite is a set of several simple unit tests. Most of the tests were written when a typo was been found, so that we hope next time it'll be detected automatically. When a problem was found, a test was created and the test failed. After the problem was fixed the test passed. The statistics section contains special tests: we are aware that they do not pass, but their actual output is interesting for us. The output of the testsuite is sent to the `frugalware-devel@` mailing list daily.

Since the tests in the testsuite section should pass, if one fails it is expected to be fixed within a day, especially if your name is listed next to a line.

You can find the tests under the `/t` directory of FST, the statistics are under `/t/s`.

34.2 Rules

Basically there are 3 simple rules for these tests:

- If the first argument is `--help`, they should print a short (less than 80 chars) description. This will be displayed if the test fails as sometimes the name of the test may not be descriptive enough.
- The tests are called in a `./testname` form, without any argument. This allows you to use various interpreted programming languages (python, bash, etc.).
- If the test *passes*, there should be no output. This means that there may be a `-v` or `--verbose` option to generate output even if the test passes, that's not a problem. If the test *fails* there must be some output. For example if there are problematic packages, then it's recommended to list each package in a separate line with their path under FST.

34.3 Technical details

Given that all the files in the fdb and fpm files are owned by root, if you want to operate on them, then you need to use fakeroot. The testsuite wrapper won't do this for you. A common practice is to write a generic python script that operates on the fdb, then create a shell wrapper for each arch, which will call the python script via fakeroot.

35 Translations

35.1 Introduction

Localization is important for every user who doesn't speak English fluently. If your native language is not English, then you can help us by translating a few sentences to your native language. If you would like to help, the following steps are necessary:

- Visit the web interface and register.
- Select your language (ie. if you would like to contribute French translation, select French). If your language is not listed, then ask for addition on our developer mailing list.
- Select what projects you would like to translate. It's good to start with some smaller project like the homepage or the setup. If the given project has no .po file for your language, contact us.
- Now you can begin translating, but your changes won't hit the master repo, you need additional permissions to commit from the sandbox. Ask us for commit access.

A few tips if you're new to pootle:

- By default you can edit the whole translation, but usually you would like to see only the untranslated and fuzzy strings. You can search for them by clicking on "Show editing functions" then selecting "Quick translate".
- You can commit a po file by clicking on "Show editing functions" then selecting "Commit".
- You can search for fuzzy translations by clicking on "Show editing functions", selecting "Show checks" and then the "isfuzzy" check.

35.2 Rules

There are not many, at the moment.

- Please don't translate the `==NAME` and `==SYNOPSIS` strings in the manpages, docbook does it already and asciidoc fails to create the manpage if it's already translated.
- The first translator for a language (this can be changed if requested) receives all rights for a given project, except: Suggest, Overwrite, Assign, Administrate.

35.3 Goals

When we created the current mechanism of handling translations, we had the following goals:

- When we modify source code or documentation, the translators should be able to begin the necessary (if any) translations without any manual action.
- It would be be nice to overview the localization status of a language.
- It should be easy to maintain the translation (ie. doing a manual sync for big documents is rather problematic).
- Translators are not developers, write access to the translations should not require any other access right.
- It should be possible for anyone to translate, but only given users should be able to push changes.

35.4 Overview

Now let's see how all this is possible. We'll take our `asciidoc` documentation as an example.

First, we need to extract the translatable strings from the sources. This is an important step since this way a document is split into paragraphs and you can then later translate even a single paragraph rather then choosing between translating a 10-page-length document entirely or not. We use `po4a` for this purpose. It creates a template, named `docs.pot`, which is transferred daily to the translation server.

(For source codes we usually use the `intltool-update` utility to extract translatable strings.)

Right after the transfer, the po localization files are updated using `msgmerge` from the `gettext` package: this way the translators do not have to re-translate the strings which are already done.

On that machine, we use a web interface for the translation. This has several advantages:

- The translators can register and begin their work without any confirmation from our developer team.
- Those accounts are - of course - not real unix accounts but just virtual ones.
- We can give commit access for users by specifying their project and language. So everybody can make translations but only users we know can push the changes.
- Collaboration for people who do not know what a patch or a version control system is now should not be a big problem. This is important since for example the whole documentation is one big file per language.

Once a user with enough privileges pushes the translation to our `git` version control system, we can use it. The documentation is built daily and we pull the new translations from the dedicated repo before each build.

The output of the English build is available here. If it contains any error or warning, the testsuite will let us know by including them in the daily testsuite mail, sent to the developer mailing list. The log of the localization builds is available here.

There we use `po4a` again to reconstruct the original (now in some language other than English) document from the translated strings.

For source code we pull the translations right before creating a release tarball so. This has the following benefits:

- We ship the latest translations
- Once the tarball is ready, users who would like to compile the source code should not fetch the translations manually.

The proof of concept for this mechanism is our French documentation which is more than 80 pages length and includes zero percent of manual editing by the developers (while till now we had to push the submitted - by email and other undocumented channels - translated documents manually, hoping that the newer version is better than the old was).

36 How to port Frugalware to a new architecture

36.1 Introduction

This document is a draft about how to port Frugalware to a new architecture.

36.2 Toolchain

- Install any existing distro to the given architecture. No matter what kind of it, but make sure you install the normal development tools like header files, gcc, make, etc.
- Compile from source (based on the FrugalBuilds) our development tools like pacman-g2, pacman-tools (+ deps: libarchive, etc if they are not available.)
- Build a minimal toolchain: binutils, gcc, glibc (in this order) outside chroot, with dep checking disabled (makepkg -dHcu).
- Build packages which are necessary to build in chroot: see the COREPKGS variable in /etc/makepkg.conf (same makepkg switches).

Given that repoman won't allow you to upload which are not built in chroot, here is a simple script to upload and register then till you don't have a chroot:

```
#!/bin/sh
scp *.fpm genesis:git/current/frugalware-NEWARCH
pkgname=$(pwd|sed 's|.*/||')
ssh genesis "cd git/current/frugalware-NEWARCH; arch=NEWARCH updatesync upd frugalware- ↩
    current.fdb ../source/*/$pkgname/FrugalBuild"
```

Replace genesis with the server name and git/current with an other path if you don't have such a symlink in your HOME.

Now you can start building in chroot and uploading real packages.

Note
Yes, this means that you have to build the toolchain twice. Also known as *bootstrapping*.

36.3 Base system

You should start porting with packages from the *base* category, once you are done with it, you should be able to install (manually) a bootable system, after manually configuring a boot manager.

36.4 The rest

That depends on your needs, you can port additional packages as well.

37 GNU Free Documentation License

Version 1.2, November 2002

```
Copyright (C) 2000,2001,2002  Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA  02110-1301  USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
```

37.1 PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

37.2 APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned

below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

37.3 VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

37.4 COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

37.5 MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

37.6 COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or

else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

37.7 COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

37.8 AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

37.9 TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

37.10 TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document 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.

37.11 FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation 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. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.