

Release Notes for Debian GNU/Linux 6.0 (squeeze), PowerPC

The Debian Documentation Project (<http://www.debian.org/doc/>)

May 3, 2013

Release Notes for Debian GNU/Linux 6.0 (squeeze), PowerPC

Published February 4th, 2011

This document is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License, version 2, as published by the Free Software Foundation.

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

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

The license text can also be found at <http://www.gnu.org/licenses/gpl-2.0.html> and `/usr/share/common-licenses/GPL-2` on Debian GNU/Linux.

Contents

1	Introduction	1
1.1	Reporting bugs on this document	1
1.2	Contributing upgrade reports	1
1.3	Sources for this document	2
2	What's new in Debian GNU/Linux 6.0	3
2.1	What's new in the distribution?	3
2.1.1	CDs, DVDs and BDs	4
2.1.2	Firmware moved to the non-free section	4
2.1.3	Package management	5
2.1.4	Dependency booting	5
2.1.5	Unified keyboard settings	5
2.1.6	Kernel mode setting	5
2.1.7	LDAP support	6
2.1.8	The stable-updates section	6
2.1.9	backports.org/backports.debian.org	6
2.2	Comprehensive support for neuroimaging research	6
3	Installation System	7
3.1	What's new in the installation system?	7
3.1.1	Major changes	7
3.1.2	Automated installation	8
4	Upgrades from Debian 5.0 (lenny)	9
4.1	Preparing for the upgrade	9
4.1.1	Back up any data or configuration information	9
4.1.2	Inform users in advance	9
4.1.3	Prepare for downtime on services	9
4.1.4	Prepare for recovery	10
4.1.4.1	Debug shell during boot using initrd	10
4.1.5	Prepare a safe environment for the upgrade	10
4.1.6	Remove conflicting packages	11
4.2	Checking system status	11
4.2.1	Review actions pending in package manager	11
4.2.2	Disabling APT pinning	12
4.2.3	Checking packages status	12
4.2.4	The proposed-updates section	12
4.2.5	Unofficial sources and backports	12
4.3	Preparing sources for APT	13
4.3.1	Adding APT Internet sources	13
4.3.2	Adding APT sources for a local mirror	13
4.3.3	Adding APT source from CD-ROM or DVD	13
4.4	Upgrading packages	14
4.4.1	Recording the session	14
4.4.2	Updating the package list	15
4.4.3	Make sure you have sufficient space for the upgrade	15
4.4.4	Minimal system upgrade	16
4.4.5	Upgrading the kernel and udev	17
4.4.6	Upgrading the system	17
4.5	Possible issues during upgrade	18
4.5.1	cryptoloop support not included in the squeeze Linux kernel	18
4.5.2	Expected removals	18
4.5.3	Errors running aptitude or apt-get	18
4.5.4	Conflicts or Pre-Depends loops	18

4.5.5	File conflicts	18
4.5.6	Configuration changes	19
4.5.7	Change of session to console	19
4.5.8	Special care for specific packages	19
4.5.8.1	Evolution	19
4.6	Upgrading your kernel and related packages	20
4.6.1	Installing the kernel metapackage	20
4.6.2	Device enumeration reordering	20
4.6.3	Boot timing issues (waiting for root device)	21
4.7	Preparing for the next release	21
4.8	Deprecated components	21
4.9	Obsolete packages	21
4.9.1	Dummy packages	23
5	Issues to be aware of for squeeze	25
5.1	Potential problems	25
5.1.1	Migration of disk drivers from IDE to PATA subsystem	25
5.1.2	mdadm metadata format change requires recent Grub	25
5.1.3	pam_userdb.so breakage with newer libdb	25
5.1.4	Potential issues with diversions of /bin/sh	25
5.1.5	Change in kernel policy regarding resource conflicts	26
5.2	LDAP support	26
5.3	sieve service moving to its IANA-allocated port	26
5.4	Security status of web browsers	27
5.5	KDE desktop	27
5.5.1	Upgrading from KDE 3	27
5.5.2	New KDE metapackages	27
5.6	GNOME desktop changes and support	28
5.6.1	GDM 2.20 and 2.30	28
5.6.2	Device and other administrative permissions	28
5.6.3	network-manager and ifupdown interaction	28
5.7	Graphics stack changes	28
5.7.1	Obsolete Xorg drivers	29
5.7.2	Kernel mode setting	29
5.7.3	Input device hotplug	29
5.7.4	X server “zapping”	29
5.8	Munin web path change	29
5.9	Shorewall upgrade instructions	29
6	More information on Debian GNU/Linux	31
6.1	Further reading	31
6.2	Getting help	31
6.2.1	Mailing lists	31
6.2.2	Internet Relay Chat	31
6.3	Reporting bugs	31
6.4	Contributing to Debian	32
7	Glossary	33
A	Managing your lenny system before the upgrade	35
A.1	Upgrading your lenny system	35
A.2	Checking your sources list	35
A.3	Upgrade legacy locales to UTF-8	36
B	Contributors to the Release Notes	37
	Index	39

Chapter 1

Introduction

This document informs users of the Debian GNU/Linux distribution about major changes in version 6.0 (codenamed squeeze).

The release notes provide information on how to upgrade safely from release 5.0 (codenamed lenny) to the current release and inform users of known potential issues they could encounter in that process.

You can get the most recent version of this document from <http://www.debian.org/releases/squeeze/releasenotes>. If in doubt, check the date on the first page to make sure you are reading a current version.

Caution



Note that it is impossible to list every known issue and that therefore a selection has been made based on a combination of the expected prevalence and impact of issues.

Please note that we only support and document upgrading from the previous release of Debian (in this case, the upgrade from 5.0). If you need to upgrade from older releases, we suggest you read previous editions of the release notes and upgrade to 5.0 first.

1.1 Reporting bugs on this document

We have attempted to test all the different upgrade steps described in this document and to anticipate all the possible issues our users might encounter.

Nevertheless, if you think you have found a bug (incorrect information or information that is missing) in this documentation, please file a bug in the [bug tracking system](http://bugs.debian.org/) (<http://bugs.debian.org/>) against the `release-notes` package. You might want to review first the [existing bug reports](http://bugs.debian.org/release-notes) (<http://bugs.debian.org/release-notes>) in case the issue you've found has already been reported. Feel free to add additional information to existing bug reports if you can contribute content for this document.

We appreciate, and encourage, reports providing patches to the document's sources. You will find more information describing how to obtain the sources of this document in [Section 1.3](#).

1.2 Contributing upgrade reports

We welcome any information from users related to upgrades from lenny to squeeze. If you are willing to share information please file a bug in the [bug tracking system](http://bugs.debian.org/) (<http://bugs.debian.org/>) against the `upgrade-reports` package with your results. We request that you compress any attachments that are included (using `gzip`).

Please include the following information when submitting your upgrade report:

- The status of your package database before and after the upgrade: `dpkg`'s status database available at `/var/lib/dpkg/status` and `apt`'s package state information, available at `/var/lib/`

`apt/extended_states`. You should have made a backup before the upgrade as described at Section 4.1.1, but you can also find backups of `/var/lib/dpkg/status` in `/var/backups`.

- Session logs created using **script**, as described in Section 4.4.1.
- Your `apt` logs, available at `/var/log/apt/term.log` or your **aptitude** logs, available at `/var/log/aptitude`.

Note



You should take some time to review and remove any sensitive and/or confidential information from the logs before including them in a bug report as the information will be published in a public database.

1.3 Sources for this document

The source of this document is in DocBook XML format. The HTML version is generated using `docbook-xsl` and `xsltproc`. The PDF version is generated using `dblatex` or `xmlroff`. Sources for the Release Notes are available in the SVN repository of the *Debian Documentation Project*. You can use the **web interface** (<http://svn.debian.org/viewsvn/ddp/manuals/trunk/release-notes/>) to access its files individually through the web and see their changes. For more information on how to access the SVN please consult the **Debian Documentation Project SVN information pages** (<http://www.debian.org/doc/cvs>).

Chapter 2

What's new in Debian GNU/Linux 6.0

The [Wiki](http://wiki.debian.org/NewInSqueeze) (<http://wiki.debian.org/NewInSqueeze>) has more information about this topic.

This release drops official support for the [HP PA-RISC \('hppa'\)](http://lists.debian.org/debian-devel-announce/2010/09/msg00008.html) (<http://lists.debian.org/debian-devel-announce/2010/09/msg00008.html>), Alpha ('alpha') and ARM ('arm') architectures.

The following are the officially supported architectures for Debian GNU/Linux squeeze:

- 32-bit PC ('i386')
- SPARC ('sparc')
- PowerPC ('powerpc')
- MIPS ('mips' (big-endian) and 'mipsel' (little-endian))
- Intel Itanium ('ia64')
- S/390 ('s390')
- 64-bit PC ('amd64')
- ARM EABI ('armel')

In addition to the officially supported architectures, Debian GNU/Linux squeeze introduces the GNU/kFreeBSD ports ('kfreebsd-amd64' and 'kfreebsd-i386') as a technology preview. These ports are the first ones included in a Debian release which aren't based on the Linux kernel, but instead use the FreeBSD kernel with a GNU userland. Users of these versions however should be warned that the quality of these ports is still catching up with the outstanding high quality of our Linux ports, and that some advanced desktop features are not supported yet. However, the support of common server software is strong and extends the features of Linux-based Debian versions by the unique features known from the BSD world. This is the first time a Linux distribution has been extended to also allow use of a non-Linux kernel.

You can read more about port status, and port-specific information for your architecture at the [Debian port web pages](http://www.debian.org/ports/) (<http://www.debian.org/ports/>).

2.1 What's new in the distribution?

This new release of Debian again comes with a lot more software than its predecessor lenny; the distribution includes over 10352 new packages, for a total of over 29050 packages. Most of the software in the distribution has been updated: over 15436 software packages (this is 67% of all packages in lenny). Also, a significant number of packages (over 4238, 18% of the packages in lenny) have for various reasons been removed from the distribution. You will not see any updates for these packages and they will be marked as 'obsolete' in package management front-ends.

With this release, Debian GNU/Linux updates from X.Org 7.3 to X.Org 7.5.

Debian GNU/Linux again ships with several desktop applications and environments. Among others it now includes the desktop environments GNOME 2.30¹, KDE 4.4.5, Xfce 4.6.2, and LXDE 0.5.0. Productivity applications have also been upgraded, including the office suites OpenOffice.org 3.2.1 and KOffice 2.2.1 as well as GNUcash 2.2.9, GNUmeric 1.10.8 and Abiword 2.8.2.

Updates of other desktop applications include the upgrade to Evolution 2.30.3 and Pidgin 2.7.3. The Mozilla suite has also been updated: `iceweasel` (version 3.5.13) is the unbranded Firefox web browser and `icedove` (version 3.0.7) is the unbranded Thunderbird mail client.

Among many others, this release also includes the following software updates:

Package	Version in 5.0 (lenny)	Version in 6.0 (squeeze)
Apache	2.2.9	2.2.16
BIND DNS Server	9.6.0	9.7.1
Cherokee web server	0.7.2	1.0.8
Courier MTA	0.60.0	0.63.0
Dia	0.96.1	0.97.1
Ekiga VoIP Client	2.0.12	3.2.7
Exim default email server	4.69	4.72
GNU Compiler Collection as default compiler	4.3.2	4.4.5
GIMP	2.4.7	2.6.10
the GNU C library	2.7	2.11.2
lighttpd	1.4.19	1.4.28
maradns	1.3.07.09	1.4.03
MySQL	5.0.51a	5.1.49
OpenLDAP	2.4.11	2.4.23
OpenSSH	5.1p1	5.5p1
PHP	5.2.6	5.3.2
Postfix MTA	2.5.5	2.7.1
PostgreSQL	8.3.5	8.4.5
Python	2.5.2	2.6.6
Samba	3.2.5	3.5.5
Tomcat	5.5.26	6.0.28

Debian still supports Linux Standard Base (LSB) version 3.2.

2.1.1 CDs, DVDs and BDs

The official Debian GNU/Linux distribution now ships on 7 to 8 binary DVDs or 44 to 53 binary CDs (depending on the architecture) and 6 source DVDs or 33 source CDs. Additionally, there is a *multi-arch* DVD, with a subset of the release for the `amd64` and `i386` architectures, along with the source code. Debian GNU/Linux is also released as Blu-ray (BD) images, 2 each for the `amd64` and `i386` architectures, or one for the source code. For size reasons, some very large packages are omitted from the CD builds; these packages fit better in the DVD and BD builds, so are still included there.

A new feature with squeeze is the addition of isohybrid support to the `i386` and `amd64` CDs, DVDs and BDs. To make a USB stick bootable with one of these images used to mean following some extra procedures after downloading the image; instead, now all that is required is to simply write the image directly to the USB stick. For more information please see the “Preparing Files for USB Memory Stick Booting” section in the [Installation Guide](http://www.debian.org/releases/stable/installmanual) (<http://www.debian.org/releases/stable/installmanual>).

2.1.2 Firmware moved to the non-free section

Some drivers included in the Linux kernel used to contain non-free firmware blobs. Starting from squeeze this firmware has been moved to separate packages in the non-free section of the archive, such as `firmware-linux`. If such packages are installed, the firmware will be loaded automatically when required.

¹ With some modules from GNOME 2.32.

2.1.3 Package management

The preferred program for interactive package management from a terminal is **aptitude**. For a non-interactive command line interface for package management, it is recommended to use **apt-get**. **apt-get** is also the preferred tool for upgrades between major releases. If you are still using **dselect**, you should switch to **aptitude** as the official front-end for package management.

For squeeze APT automatically installs recommended packages by default². This can be changed by adding the following line in `/etc/apt/apt.conf`:

```
APT::Install-Recommends "false";
```

2.1.4 Dependency booting

An important improvement in the Debian GNU/Linux boot system is the introduction of dependency-based boot sequencing and parallel boot. This feature is enabled by default in new installs and it will be enabled for upgrades from lenny, if possible.

This feature is enabled through the use of `insserv` by `sysv-rc` to order `init.d` scripts based on their declared dependencies³. It has been possible after a sustained effort to adapt all the boot scripts of packages provided in the distribution as well as the boot system itself.

With dependency-based boot sequencing it is also now possible to run the boot system scripts in parallel which can, under most circumstances, improve the speed of the boot process. This feature is enabled by default, in new systems and upgrades, whenever possible. To disable it specify

```
CONCURRENCY=none
```

in `/etc/default/rcS`. For more information on this feature refer to the information available in `/usr/share/doc/insserv/README.Debian`.

2.1.5 Unified keyboard settings

In this new release, the settings for the keyboard have been unified so that both the console and the Xorg server use the same settings. The keyboard settings are now defined in the `/etc/default/keyboard` configuration file which overrides the keyboard defined in Xorg's configuration file.

The `console-setup` package now handles the keyboard for both environments as well as the font configuration for the console. You can reconfigure the keyboard layout and related settings by executing **`dpkg-reconfigure keyboard-configuration`** or by manually editing the `/etc/default/keyboard` configuration file.

2.1.6 Kernel mode setting

Graphics mode setting code for the most common desktop chipsets (from Intel, ATI/AMD and NVIDIA) has moved from the respective Xorg drivers to the Linux kernel. This provides a number of advantages, such as:

- More reliable suspend and resume
- Ability to use graphics devices without X
- Faster VT switch
- Native mode text console

More details are in Section 5.7 and in [the Debian wiki](http://wiki.debian.org/KernelModesetting) (<http://wiki.debian.org/KernelModesetting>).

² This change implies that disk requirements for tasks selected through the debian installer have increased too. For more information please see the "Disk Space Needed for Tasks" chapter in the [Installation Guide](http://www.debian.org/releases/stable/installmanual) (<http://www.debian.org/releases/stable/installmanual>).

³ These dependences are declared through the use of the header format specified in the Linux Standard Base (LSB)

2.1.7 LDAP support

This Debian release comes with several options for implementing client-side authentication using LDAP. Users of the `libnss-ldap` and `libpam-ldap` packages should consider upgrading to `libnss-ldapd` and `libpam-ldapd`.

These newer packages delegate the LDAP queries to a central unprivileged daemon (`nslcd`) that provides separation between the process using the LDAP information and the daemon performing LDAP queries. This simplifies handling of secured LDAP connections, LDAP authentication credentials, provides a simpler mechanism to perform connection fail-over and debugging and avoids loading LDAP and related libraries into most applications.

Upgrading to `libnss-ldapd` and `libpam-ldapd` should be easy as existing configuration information will be mostly reused. Only for advanced configuration should any manual reconfiguration be necessary.

These packages however currently lack support for nested groups and only support password change using the LDAP password modify EXOP operation.

2.1.8 The stable-updates section

Some packages from `proposed-updates` may also be made available via the `squeeze-updates` mechanism. This path will be used for updates which many users may wish to install on their systems before the next point release is made, such as updates to virus scanners and timezone data. All packages from `squeeze-updates` will be included in point releases.

Note that this replaces the functionality previously provided by the [volatile.debian.org archive](http://volatile.debian.org/) (<http://volatile.debian.org/>).

In order to use packages from `squeeze-updates`, you can add an entry to your `sources.list`:

```
deb      http://mirrors.kernel.org/debian squeeze-updates main contrib
deb-src  http://mirrors.kernel.org/debian squeeze-updates main contrib
```

The next time you run `apt-get update`, the system will become aware of the packages in the `squeeze-updates` section and will consider them when looking for packages to upgrade.

Note that if `APT::Default-Release` is set in your `/etc/apt/apt.conf` (or in any of `/etc/apt/apt.conf.d/*`), then, in order for automatic upgrades to work, it is necessary to add the following configuration block into `/etc/apt/preferences` (see `apt_preferences(5)` for more information):

```
Package: *
Pin: release o=Debian GNU/Linux,n=squeeze-updates
Pin-Priority: 990
```

When a new package is made available via `squeeze-updates`, this will be announced on the [debian-stable-announce](http://lists.debian.org/debian-stable-announce/) (<http://lists.debian.org/debian-stable-announce/>) mailing list.

2.1.9 backports.org/backports.debian.org

The service provided by the `backports.org` repositories has been integrated in Debian infrastructure and is now an official Debian service (<http://www.debian.org/News/2010/20100905>), hosted at backports.debian.org (<http://backports.debian.org/>).

2.2 Comprehensive support for neuroimaging research

Debian GNU/Linux 6.0 is the first GNU/Linux distribution release ever to offer comprehensive support for magnetic resonance imaging (MRI) based neuroimaging research. It comes with up-to-date software for structural image analysis (e.g. `ants`), diffusion imaging and tractography (e.g. `mrtrix`), stimulus delivery (e.g. `psychopy`), MRI sequence development (e.g. `odin`), as well as a number of versatile data processing and analysis suites (e.g. `nipype`). Moreover, this release has built-in support for all major neuroimaging data formats. See the [Debian Science](http://blends.alioth.debian.org/science/tasks/neuroscience-cognitive) (<http://blends.alioth.debian.org/science/tasks/neuroscience-cognitive>) and [Debian Med](http://debian-med.alioth.debian.org/tasks/imaging) (<http://debian-med.alioth.debian.org/tasks/imaging>) task pages for a comprehensive list of included software and the [NeuroDebian webpage](http://neuro.debian.net) (<http://neuro.debian.net>) for further information.

Chapter 3

Installation System

The Debian Installer is the official installation system for Debian. It offers a variety of installation methods. Which methods are available to install your system depends on your architecture.

Images of the installer for squeeze can be found together with the Installation Guide on the [Debian website](http://www.debian.org/releases/stable/debian-installer/) (<http://www.debian.org/releases/stable/debian-installer/>).

The Installation Guide is also included on the first CD/DVD of the official Debian CD/DVD sets, at:

```
/doc/install/manual/language/index.html
```

You may also want to check the [errata](http://www.debian.org/releases/stable/debian-installer/index#errata) (<http://www.debian.org/releases/stable/debian-installer/index#errata>) for debian-installer for a list of known issues.

3.1 What's new in the installation system?

There has been a lot of development on the Debian Installer since its first official release with Debian GNU/Linux 3.1 (sarge) resulting in both improved hardware support and some exciting new features.

In these Release Notes we'll only list the major changes in the installer. If you are interested in an overview of the detailed changes since lenny, please check the release announcements for the squeeze beta and RC releases available from the Debian Installer's [news history](http://www.debian.org/devel/debian-installer/News/) (<http://www.debian.org/devel/debian-installer/News/>).

3.1.1 Major changes

Dropped platforms Support for the Alpha ('alpha'), ARM ('arm') and HP PA-RISC ('hppa') architectures has been dropped from the installer. The 'arm' architecture is obsoleted by the ARM EABI ('armel') port.

Help during the installation process The dialogs presented during the installation process now provide help information. Although not currently used in all dialogs, this feature would be increasingly used in future releases. This will improve the user experience during the installation process, especially for new users.

Installation of Recommended packages The installation system will install all recommended packages by default throughout the process except for some specific situations in which the general setting gives undesired results.

Automatic installation of hardware-specific packages The system will automatically select for installation hardware-specific packages when they are appropriate. This is achieved through the use of `discover-pkginstall` from the `discover` package.

Support for installation of previous releases The installation system can be also used for the installation of previous release, such as lenny.

Improved mirror selection The installation system provides better support for installing both squeeze as well as lenny and older releases (through the use of `archive.debian.org`). In addition, it will also check that the selected mirror is consistent and holds the selected release.

Changes in partitioning features This release of the installer supports the use of the ext4 file system and it also simplifies the creation of RAID, LVM and crypto protected partitioning systems. Support for the reiserfs file system is no longer included by default, although it can be optionally loaded.

Support for loading firmware debs during installation It is now possible to load firmware package files from the installation media in addition to removable media, allowing the creation of PXE images and CDs/DVDs with included firmware packages.

Starting with Debian 6.0, non-free firmware has been moved out of main. To install Debian on hardware that needs non-free firmware, you can either provide the firmware yourself during installation or use pre-made non-free CDs/DVDs which include the firmware. See the [Getting Debian section](#) (<http://www.debian.org/distrib>) on the Debian website for more information.

New languages Thanks to the huge efforts of translators, Debian GNU/Linux can now be installed in 70 languages. This is seven more languages than in lenny. Most languages are available in both the text-based installation user interface and the graphical user interface, while some are only available in the graphical user interface.

Languages added in this release include:

- Asturian, Estonian, Icelandic, Kazakh and Persian have been added to the graphical and text-based installer.
- Kannada, Lao, Sinhala and Telugu have been added to the graphical installer.
- Thai, previously available only in the graphical user interface, is now available also in the text-based installation user interface too.

Due to the lack of translation updates two languages were dropped in this release: Wolof and Welsh.

Improved localisation selection The selection of localisation-related values (language, location and locale settings) is now less interdependent and more flexible. Users will be able to customize the system to their localisation needs more easily while still make it comfortable to use for users that want to select the locale most common for the country they reside in.

Additionally, the consequences of localisation choices (such as timezone, keymap and mirror selection) are now more obvious to the user.

Live system installation The installer now supports live systems in two ways. First, an installer included on live system media can use the contents of the live system in place of the regular installation of the base system. Second, the installer may now be launched while running the live system, allowing the user to do other things with the live system during the install. Both features are built into the Debian Live images offered at <http://cdimage.debian.org/>.

3.1.2 Automated installation

Some changes mentioned in the previous section also imply changes in the support in the installer for automated installation using preconfiguration files. This means that if you have existing preconfiguration files that worked with the lenny installer, you cannot expect these to work with the new installer without modification.

The [Installation Guide](#) (<http://www.debian.org/releases/stable/installmanual>) has an updated separate appendix with extensive documentation on using preconfiguration.

Chapter 4

Upgrades from Debian 5.0 (lenny)

4.1 Preparing for the upgrade

We suggest that before upgrading you also read the information in Chapter 5. That chapter covers potential issues not directly related to the upgrade process but which could still be important to know about before you begin.

4.1.1 Back up any data or configuration information

Before upgrading your system, it is strongly recommended that you make a full backup, or at least back up any data or configuration information you can't afford to lose. The upgrade tools and process are quite reliable, but a hardware failure in the middle of an upgrade could result in a severely damaged system.

The main things you'll want to back up are the contents of `/etc`, `/var/lib/dpkg`, `/var/lib/apt/extended_states` and the output of `dpkg --get-selections "*" (the quotes are important)`. If you use **aptitude** to manage packages on your system, you will also want to back up `/var/lib/aptitude/pkgstates`.

The upgrade process itself does not modify anything in the `/home` directory. However, some applications (e.g. parts of the Mozilla suite, and the GNOME and KDE desktop environments) are known to overwrite existing user settings with new defaults when a new version of the application is first started by a user. As a precaution, you may want to make a backup of the hidden files and directories (“dot-files”) in users' home directories. This backup may help to restore or recreate the old settings. You may also want to inform users about this.

Any package installation operation must be run with superuser privileges, so either log in as `root` or use **su** or **sudo** to gain the necessary access rights.

The upgrade has a few preconditions; you should check them before actually executing the upgrade.

4.1.2 Inform users in advance

It's wise to inform all users in advance of any upgrades you're planning, although users accessing your system via an **ssh** connection should notice little during the upgrade, and should be able to continue working.

If you wish to take extra precautions, back up or unmount the `/home` partition before upgrading.

You will have to do a kernel upgrade when upgrading to squeeze, so a reboot will be necessary.

4.1.3 Prepare for downtime on services

There might be services that are offered by the system which are associated with packages that will be included in the upgrade. If this is the case, please note that, during the upgrade, these services will be stopped while their associated packages are being replaced and configured. During this time, these services will not be available.

The precise downtime for these services will vary depending on the number of packages being upgraded in the system, and it also includes the time the system administrator answers the configuration questions from different package upgrades (if any). Notice that if the upgrade process is left unattended

and the system requests input throughout the upgrade there is a high possibility of services being unavailable¹ for a significant period of time.

If the system being upgraded provides critical services for your users or the network², you can reduce the downtime if you do a minimal system upgrade, as described in Section 4.4.4, followed by a kernel upgrade and reboot (see Section 4.4.5), and then upgrade the packages associated with your critical services. Upgrade these packages prior to doing the full upgrade described in Section 4.4.6. This way you can ensure that these critical services are running and available through the full upgrade process, and their downtime is reduced.

4.1.4 Prepare for recovery

Because of the many changes in the kernel between lenny and squeeze regarding drivers, hardware discovery and the naming and ordering of device files, there is a real risk that you may experience problems rebooting your system after the upgrade. A lot of known potential issues are documented in this and the next chapters of these Release Notes.

For that reason it makes sense to ensure that you will be able to recover if your system should fail to reboot or, for remotely managed systems, fail to bring up networking.

If you are upgrading remotely via an `ssh` link it is highly recommended that you take the necessary precautions to be able to access the server through a remote serial terminal. There is a chance that, after upgrading the kernel and rebooting, some devices will be renamed (as described in Section 4.6.2) and you will have to fix the system configuration through a local console. Also, if the system is rebooted accidentally in the middle of an upgrade there is a chance you will need to recover using a local console.

The most obvious thing to try first is to reboot with your old kernel. However, for various reasons documented elsewhere in this document, this is not guaranteed to work.

If that fails, you will need an alternative way to boot your system so you can access and repair it. One option is to use a special rescue image or a Linux live CD. After booting from that, you should be able to mount your root file system and `chroot` into it to investigate and fix the problem.

Another option we'd like to recommend is to use the *rescue mode* of the squeeze Debian Installer. The advantage of using the installer is that you can choose between its many installation methods for one that best suits your situation. For more information, please consult the section "Recovering a Broken System" in chapter 8 of the [Installation Guide](http://www.debian.org/releases/stable/installmanual) (<http://www.debian.org/releases/stable/installmanual>) and the [Debian Installer FAQ](http://wiki.debian.org/DebianInstaller/FAQ) (<http://wiki.debian.org/DebianInstaller/FAQ>).

4.1.4.1 Debug shell during boot using `initrd`

The `initramfs-tools` includes a debug shell³ in the `initrd`s it generates. If for example the `initrd` is unable to mount your root file system, you will be dropped into this debug shell which has basic commands available to help trace the problem and possibly fix it.

Basic things to check are: presence of correct device files in `/dev`; what modules are loaded (`cat /proc/modules`); output of `dmesg` for errors loading drivers. The output of `dmesg` will also show what device files have been assigned to which disks; you should check that against the output of `echo $ROOT` to make sure that the root file system is on the expected device.

If you do manage to fix the problem, typing `exit` will quit the debug shell and continue the boot process at the point it failed. Of course you will also need to fix the underlying problem and regenerate the `initrd` so the next boot won't fail again.

4.1.5 Prepare a safe environment for the upgrade

The distribution upgrade should be done either locally from a textmode virtual console (or a directly connected serial terminal), or remotely via an `ssh` link.

¹ If the `debconf` priority is set to a very high level you might prevent configuration prompts, but services that rely on default answers that are not applicable to your system will fail to start.

² For example: DNS or DHCP services, specially when there is no redundancy or failover. In the DHCP case end-users might be disconnected from the network if the lease time is lower than the time it takes for the upgrade process to complete.

³ This feature can be disabled by adding the parameter `panic=0` to your boot parameters.

Important

If you are using some VPN services (such as `tinc`) they might not be available throughout the upgrade process. Please see Section [4.1.3](#).

In order to gain extra safety margin when upgrading remotely, we suggest that you run upgrade processes in the virtual console provided by the `screen` program, which enables safe reconnection and ensures the upgrade process is not interrupted even if the remote connection process fails.

Important

You should *not* upgrade using `telnet`, `rlogin`, `rsh`, or from an X session managed by `xdm`, `gdm` or `kdm` etc on the machine you are upgrading. That is because each of those services may well be terminated during the upgrade, which can result in an *inaccessible* system that is only half-upgraded. Use of the GNOME application **update-manager** is *strongly discouraged* for upgrades to new releases, as this tool relies on the desktop session remaining active.

4.1.6 Remove conflicting packages

Due to bug [#512951](http://bugs.debian.org/512951) (<http://bugs.debian.org/512951>), the `splashy` package needs to be purged prior to the upgrade.

```
# apt-get purge splashy
```

4.2 Checking system status

The upgrade process described in this chapter has been designed for upgrades from “pure” lenny systems without third-party packages. For the greatest reliability of the upgrade process, you may wish to remove third-party packages from your system before you begin upgrading.

Direct upgrades from Debian releases older than 5.0 (lenny) are not supported. Please follow the instructions in the [Release Notes for Debian GNU/Linux 5.0](http://www.debian.org/releases/lenny/releasenotes) (<http://www.debian.org/releases/lenny/releasenotes>) to upgrade to 5.0 first.

This procedure also assumes your system has been updated to the latest point release of lenny. If you have not done this or are unsure, follow the instructions in Section [A.1](#).

4.2.1 Review actions pending in package manager

In some cases, the use of `apt-get` for installing packages instead of `aptitude` might make `aptitude` consider a package as “unused” and schedule it for removal. In general, you should make sure the system is fully up-to-date and “clean” before proceeding with the upgrade.

Because of this you should review if there are any pending actions in the package manager `aptitude`. If a package is scheduled for removal or update in the package manager, it might negatively impact the upgrade procedure. Note that correcting this is only possible if your `sources.list` still points to *lenny* and not to *stable* or *squeeze*; see Section [A.2](#).

To perform this review, launch `aptitude` in “visual mode” and press `g` (“Go”). If it shows any actions, you should review them and either fix them or implement the suggested actions. If no actions are suggested you will be presented with a message saying “No packages are scheduled to be installed, removed, or upgraded”.

4.2.2 Disabling APT pinning

If you have configured APT to install certain packages from a distribution other than stable (e.g. from testing), you may have to change your APT pinning configuration (stored in `/etc/apt/preferences`) to allow the upgrade of packages to the versions in the new stable release. Further information on APT pinning can be found in `apt_preferences(5)`.

4.2.3 Checking packages status

Regardless of the method used for upgrading, it is recommended that you check the status of all packages first, and verify that all packages are in an upgradable state. The following command will show any packages which have a status of Half-Installed or Failed-Config, and those with any error status.

```
# dpkg --audit
```

You could also inspect the state of all packages on your system using `dselect`, `aptitude`, or with commands such as

```
# dpkg -l | pager
```

or

```
# dpkg --get-selections "*" > ~/curr-pkgs.txt
```

It is desirable to remove any holds before upgrading. If any package that is essential for the upgrade is on hold, the upgrade will fail.

Note that `aptitude` uses a different method for registering packages that are on hold than `apt-get` and `dselect`. You can identify packages on hold for `aptitude` with

```
# aptitude search "~ahold"
```

If you want to check which packages you had on hold for `apt-get`, you should use

```
# dpkg --get-selections | grep hold
```

If you changed and recompiled a package locally, and didn't rename it or put an epoch in the version, you must put it on hold to prevent it from being upgraded.

The “hold” package state for `apt-get` can be changed using:

```
# echo package_name hold | dpkg --set-selections
```

Replace `hold` with `install` to unset the “hold” state.

If there is anything you need to fix, it is best to make sure your `sources.list` still refers to lenny as explained in Section [A.2](#).

4.2.4 The proposed-updates section

If you have listed the `proposed-updates` section in your `/etc/apt/sources.list` file, you should remove it from that file before attempting to upgrade your system. This is a precaution to reduce the likelihood of conflicts.

4.2.5 Unofficial sources and backports

If you have any non-Debian packages on your system, you should be aware that these may be removed during the upgrade because of conflicting dependencies. If these packages were installed by adding an extra package archive in your `/etc/apt/sources.list`, you should check if that archive also offers packages compiled for squeeze and change the source line accordingly at the same time as your source lines for Debian packages.

Some users may have unofficial backported “newer” versions of packages that *are* in Debian installed on their lenny system. Such packages are most likely to cause problems during an upgrade as they may result in file conflicts⁴. Section [4.5](#) has some information on how to deal with file conflicts if they should occur.

⁴ Debian's package management system normally does not allow a package to remove or replace a file owned by another package unless it has been defined to replace that package.

4.3 Preparing sources for APT

Before starting the upgrade you must set up apt's configuration file for package lists, `/etc/apt/sources.list`.

apt will consider all packages that can be found via any “deb” line, and install the package with the highest version number, giving priority to the first line in the file (thus where you have multiple mirror locations, you'd typically first name a local hard disk, then CD-ROMs, and then HTTP/FTP mirrors).

A release can often be referred to both by its codename (e.g. `lenny`, `squeeze`) and by its status name (i.e. `oldstable`, `stable`, `testing`, `unstable`). Referring to a release by its codename has the advantage that you will never be surprised by a new release and for this reason is the approach taken here. It does of course mean that you will have to watch out for release announcements yourself. If you use the status name instead, you will just see loads of updates for packages available as soon as a release has happened.

4.3.1 Adding APT Internet sources

The default configuration is set up for installation from main Debian Internet servers, but you may wish to modify `/etc/apt/sources.list` to use other mirrors, preferably a mirror that is network-wise closest to you.

Debian HTTP or FTP mirror addresses can be found at <http://www.debian.org/distrib/ftplist> (look at the “list of Debian mirrors” section). HTTP mirrors are generally speedier than FTP mirrors.

For example, suppose your closest Debian mirror is `http://mirrors.kernel.org`. When inspecting that mirror with a web browser or FTP program, you will notice that the main directories are organized like this:

```
http://mirrors.kernel.org/debian/dists/squeeze/main/binary-powerpc/...
http://mirrors.kernel.org/debian/dists/squeeze/contrib/binary-powerpc/...
```

To use this mirror with apt, you add this line to your `sources.list` file:

```
deb http://mirrors.kernel.org/debian squeeze main contrib
```

Note that the `dists`' is added implicitly, and the arguments after the release name are used to expand the path into multiple directories.

After adding your new sources, disable the previously existing “deb” lines in `sources.list` by placing a hash sign (#) in front of them.

4.3.2 Adding APT sources for a local mirror

Instead of using HTTP or FTP package mirrors, you may wish to modify `/etc/apt/sources.list` to use a mirror on a local disk (possibly mounted over NFS).

For example, your package mirror may be under `/var/ftp/debian/`, and have main directories like this:

```
/var/ftp/debian/dists/squeeze/main/binary-powerpc/...
/var/ftp/debian/dists/squeeze/contrib/binary-powerpc/...
```

To use this with apt, add this line to your `sources.list` file:

```
deb file:/var/ftp/debian squeeze main contrib
```

Note that the `dists`' is added implicitly, and the arguments after the release name are used to expand the path into multiple directories.

After adding your new sources, disable the previously existing “deb” lines in `sources.list` by placing a hash sign (#) in front of them.

4.3.3 Adding APT source from CD-ROM or DVD

If you want to use CDs *only*, comment out the existing “deb” lines in `/etc/apt/sources.list` by placing a hash sign (#) in front of them.

Make sure there is a line in `/etc/fstab` that enables mounting your CD-ROM drive at the `/cdrom` mount point (the exact `/cdrom` mount point is required for **apt-cdrom**). For example, if `/dev/hdc` is your CD-ROM drive, `/etc/fstab` should contain a line like:

```
/dev/hdc /cdrom auto defaults,noauto,ro 0 0
```

Note that there must be *no spaces* between the words `defaults`, `noauto`, `ro` in the fourth field. To verify it works, insert a CD and try running

```
# mount /cdrom # this will mount the CD to the mount point
# ls -aF /cdrom # this should show the CD's root directory
# umount /cdrom # this will unmount the CD
```

Next, run:

```
# apt-cdrom add
```

for each Debian Binary CD-ROM you have, to add the data about each CD to APT's database.

4.4 Upgrading packages

The recommended way to upgrade from previous Debian GNU/Linux releases is to use the package management tool **apt-get**. In previous releases, **aptitude** was recommended for this purpose, but recent versions of **apt-get** provide equivalent functionality and also have shown to more consistently give the desired upgrade results.

Don't forget to mount all needed partitions (notably the root and `/usr` partitions) read-write, with a command like:

```
# mount -o remount,rw /mountpoint
```

Next you should double-check that the APT source entries (in `/etc/apt/sources.list`) refer either to “squeeze” or to “stable”. There should not be any sources entries pointing to lenny.

Note



Source lines for a CD-ROM might sometimes refer to “unstable”; although this may be confusing, you should *not* change it.

4.4.1 Recording the session

It is strongly recommended that you use the `/usr/bin/script` program to record a transcript of the upgrade session. Then if a problem occurs, you will have a log of what happened, and if needed, can provide exact information in a bug report. To start the recording, type:

```
# script -t 2>~/upgrade-squeezestep.time -a ~/upgrade-squeezestep.script
```

or similar. If you have to rerun the typescript (e.g. if you have to reboot the system) use different `step` values to indicate which step of the upgrade you are logging. Do not put the typescript file in a temporary directory such as `/tmp` or `/var/tmp` (files in those directories may be deleted during the upgrade or during any restart).

The typescript will also allow you to review information that has scrolled off-screen. If you are at the system's console, just switch to VT2 (using `Alt+F2`) and, after logging in, use `less -R ~/root/upgrade-squeeze.script` to view the file.

After you have completed the upgrade, you can stop **script** by typing `exit` at the prompt.

If you have used the `-t` switch for **script** you can use the **scriptreplay** program to replay the whole session:

```
# scriptreplay ~/upgrade-squeeze.time ~/upgrade-squeeze.script
```

4.4.2 Updating the package list

First the list of available packages for the new release needs to be fetched. This is done by executing:

```
# apt-get update
```

4.4.3 Make sure you have sufficient space for the upgrade

You have to make sure before upgrading your system that you have sufficient hard disk space when you start the full system upgrade described in Section 4.4.6. First, any package needed for installation that is fetched from the network is stored in `/var/cache/apt/archives` (and the `partial/` subdirectory, during download), so you must make sure you have enough space on the file system partition that holds `/var/` to temporarily download the packages that will be installed in your system. After the download, you will probably need more space in other file system partitions in order to both install upgraded packages (which might contain bigger binaries or more data) and new packages that will be pulled in for the upgrade. If your system does not have sufficient space you might end up with an incomplete upgrade that might be difficult to recover from.

apt-get can show you detailed information of the disk space needed for the installation. Before executing the upgrade, you can see this estimate by running:

```
# apt-get -o APT::Get::Trivial-Only=true dist-upgrade
[ ... ]
XXX upgraded, XXX newly installed, XXX to remove and XXX not upgraded.
Need to get xx.xMB of archives.
After this operation, AAAMB of additional disk space will be used.
```

Note



Running this command at the beginning of the upgrade process may give an error, for the reasons described in the next sections. In that case you will need to wait until you've done the minimal system upgrade as in Section 4.4.4 and upgraded your kernel before running this command to estimate the disk space.

If you do not have enough space for the upgrade, **apt-get** will warn you with a message like this:

```
E: You don't have enough free space in /var/cache/apt/archives/.
```

In this situation, make sure you free up space beforehand. You can:

- Remove packages that have been previously downloaded for installation (at `/var/cache/apt/archives`). Cleaning up the package cache by running **apt-get clean** will remove all previously downloaded package files.
- Remove forgotten packages. If you have `popcon-largest-unused` installed, you can use **popcon-largest-unused** to list the packages you do not use that occupy the most space. You can also use **deborphan** or **debfostr** to find obsolete packages (see Section 4.9). Alternatively you can start **aptitude** in “visual mode” and find obsolete packages under “Obsolete and Locally Created Packages”.
- Remove packages that take up too much space and are not currently needed (you can always reinstall them after the upgrade). You can list the packages that take up the most disk space with **dpigs** (available in the `debian-goodies` package) or with **wajig** (running `wajig size`).

You can list packages that take up most of the disk space with **aptitude**. Start **aptitude** in “visual mode”, select Views → New Flat Package List, press **I** and enter `~i`, press **S** and enter `~installs size`, then it will give you nice list to work with.

- Remove translations and localization files from the system if they are not needed. You can install the `localepurge` package and configure it so that only a few selected locales are kept in the system. This will reduce the disk space consumed at `/usr/share/locale`.

- Temporarily move to another system, or permanently remove, system logs residing under `/var/log/`.
- Use a temporary `/var/cache/apt/archives`: You can use a temporary cache directory from another filesystem (USB storage device, temporary hard disk, filesystem already in use, ...)

Note



Do not use an NFS mount as the network connection could be interrupted during the upgrade.

For example, if you have a USB drive mounted on `/media/usbkey`:

1. remove the packages that have been previously downloaded for installation:

```
# apt-get clean
```

2. copy the directory `/var/cache/apt/archives` to the USB drive:

```
# cp -ax /var/cache/apt/archives /media/usbkey/
```

3. mount the temporary cache directory on the current one:

```
# mount --bind /media/usbkey/archives /var/cache/apt/archives
```

4. after the upgrade, restore the original `/var/cache/apt/archives` directory:

```
# umount /media/usbkey/archives
```

5. remove the remaining `/media/usbkey/archives`.

You can create the temporary cache directory on whatever filesystem is mounted on your system.

- Do a minimal upgrade of the system (see Section 4.4.4) or partial upgrades of the system followed by a full upgrade. This will make it possible to upgrade the system partially, and allow you to clean the package cache before the full upgrade.

Note that in order to safely remove packages, it is advisable to switch your `sources.list` back to lenny as described in Section A.2.

4.4.4 Minimal system upgrade

In some cases, doing the full upgrade (as described below) directly might remove large numbers of packages that you will want to keep. We therefore recommend a two-part upgrade process, first a minimal upgrade to overcome these conflicts, then a full upgrade as described in Section 4.4.6.

To do this first, run:

```
# apt-get upgrade
```

This has the effect of upgrading those packages which can be upgraded without requiring any other packages to be removed or installed.

The minimal system upgrade can also be useful when the system is tight on space and a full upgrade cannot be run due to space constraints.

4.4.5 Upgrading the kernel and udev

The `udev` version in `squeeze` requires a kernel of version 2.6.26 or newer with the `CONFIG_SYSFS_DEPRECATED` option disabled and the `CONFIG_INOTIFY_USER` and `CONFIG_SIGNALFD` options enabled. Because the standard Debian kernels in `lenny` (version 2.6.26) have `CONFIG_SYSFS_DEPRECATED` enabled, and the `udev` version in `lenny` will not provide all the functionality expected by the latest kernels, special care must be taken when upgrading to avoid putting your system in an unbootable state.

Booting the 2.6.26 kernel from `lenny` with the `udev` from `squeeze` may result in a failure to correctly assign names to network devices, and will also fail to apply certain additional permissions to block devices (such as access by the `disk` group). The software itself will appear to be working, but some rules (for example, network-based rules) will not be loaded properly. It is therefore strongly recommended that you upgrade the kernel on its own at this point, to ensure a compatible kernel is available before upgrading `udev`.

To proceed with this kernel upgrade, run:

```
# apt-get install linux-image-2.6-flavor
```

See Section 4.6.1 for help in determining which flavor of kernel package you should install.

The move of some firmware to separate packages in the non-free archive (see Section 2.1.2) means that it may be necessary to install additional firmware packages after upgrading to the new kernel to support some hardware. Some hardware that was operating correctly before the upgrade might fail to work once the kernel is upgraded. Look out for warning messages from the kernel install or `initramfs` generation scripts, and make sure the necessary firmware packages are installed.

Immediately after upgrading the kernel, you should also install the new `udev` to minimize the risk of other incompatibilities caused by using the old `udev` with a new kernel⁵. You can do this by running:

```
# apt-get install udev
```

You should reboot the system⁶ once you have upgraded both the kernel and `udev`.

4.4.6 Upgrading the system

Once you have taken the previous steps, you are now ready to continue with the main part of the upgrade. Execute:

```
# apt-get dist-upgrade
```

Note



The upgrade process for other releases recommended the use of **aptitude** for the upgrade. This tool is not recommended for upgrades from `lenny` to `squeeze`.

This will perform a complete upgrade of the system, i.e. install the newest available versions of all packages, and resolve all possible dependency changes between packages in different releases. If necessary, it will install some new packages (usually new library versions, or renamed packages), and remove any conflicting obsoleted packages.

When upgrading from a set of CD-ROMs (or DVDs), you will be asked to insert specific CDs at several points during the upgrade. You might have to insert the same CD multiple times; this is due to inter-related packages that have been spread out over the CDs.

New versions of currently installed packages that cannot be upgraded without changing the install status of another package will be left at their current version (displayed as “held back”). This can be resolved by either using **aptitude** to choose these packages for installation or by trying `apt-get -f install package`.

⁵ There are also known incompatibilities between the old kernel and the new `udev`. If you find issues after the reboot with the new kernel you will have to downgrade the `udev` in order to use the old one.

⁶ If you are logging the upgrade as described in Section 4.4, please, use **script** again to log the next steps of the upgrade after the reboot in order to log the result of the actions described in Section 4.4.6.

4.5 Possible issues during upgrade

The following sections describe known issues that might appear during an upgrade to squeeze.

4.5.1 cryptoloop support not included in the squeeze Linux kernel

Support for cryptoloop has been dropped from the Linux kernel packages included in Debian 6.0. Existing installations using cryptoloop need to be transitioned to dm-crypt before the upgrade.

4.5.2 Expected removals

The upgrade process to squeeze might ask for removal of packages in the system. The precise list of packages will vary depending on the set of packages that you have installed. These release notes give general advice on these removals, but if in doubt, it is recommended that you examine the package removals proposed by each method before proceeding.

Some common packages that are expected to be removed include: `autofs` (replaced by `autofs5`), `dhcp3` (replaced by `isc-dhcp`), `madwifi-source` and `python2.4` (replaced by `python2.6`). For more information about packages obsoleted in squeeze, see Section 4.9.

4.5.3 Errors running aptitude or apt-get

If an operation using `aptitude`, `apt-get`, or `dpkg` fails with the error

```
E: Dynamic MMap ran out of room
```

the default cache space is insufficient. You can solve this by either removing or commenting lines you don't need in `/etc/apt/sources.list` or increasing the cache size. The cache size can be increased by setting `APT::Cache-Limit` in `/etc/apt/apt.conf`. The following command will set it to a value that should be sufficient for the upgrade:

```
# echo 'APT::Cache-Limit "12500000";' >> /etc/apt/apt.conf
```

This assumes that you do not yet have this variable set in that file.

4.5.4 Conflicts or Pre-Depends loops

Sometimes it's necessary to enable the `APT::Force-LoopBreak` option in APT to be able to temporarily remove an essential package due to a Conflicts/Pre-Depends loop. `apt-get` will alert you of this and abort the upgrade. You can work around this by specifying the option `-o APT::Force-LoopBreak=1` on the `apt-get` command line.

It is possible that a system's dependency structure can be so corrupt as to require manual intervention. Usually this means using `apt-get` or

```
# dpkg --remove package_name
```

to eliminate some of the offending packages, or

```
# apt-get -f install
# dpkg --configure --pending
```

In extreme cases you might have to force re-installation with a command like

```
# dpkg --install /path/to/package_name.deb
```

4.5.5 File conflicts

File conflicts should not occur if you upgrade from a "pure" lenny system, but can occur if you have unofficial backports installed. A file conflict will result in an error like:

```
Unpacking <package-foo> (from <package-foo-file>) ...
dpkg: error processing <package-foo> (--install):
 trying to overwrite `<some-file-name>',
 which is also in package <package-bar>
dpkg-deb: subprocess paste killed by signal (Broken pipe)
Errors were encountered while processing:
<package-foo>
```

You can try to solve a file conflict by forcibly removing the package mentioned on the *last* line of the error message:

```
# dpkg -r --force-depends package_name
```

After fixing things up, you should be able to resume the upgrade by repeating the previously described **apt-get** commands.

4.5.6 Configuration changes

During the upgrade, you will be asked questions regarding the configuration or re-configuration of several packages. When you are asked if any file in the `/etc/init.d` directory, or the `/etc/manpath.config` file should be replaced by the package maintainer's version, it's usually necessary to answer 'yes' to ensure system consistency. You can always revert to the old versions, since they will be saved with a `.dpkg-old` extension.

If you're not sure what to do, write down the name of the package or file and sort things out at a later time. You can search in the typescript file to review the information that was on the screen during the upgrade.

4.5.7 Change of session to console

If you are running the upgrade using the system's local console you might find that at some points during the upgrade the console is shifted over to a different view and you lose visibility of the upgrade process. For example, this will happen in desktop systems when **gdm** is restarted.

To recover the console where the upgrade was running you will have to use `Ctrl+Alt+F1` to switch back to the virtual terminal 1 if in the graphical startup screen or use `Alt+F1` if in the local text-mode console. Replace F1 with the function key with the same number of the virtual terminal the upgrade was running in. You can also use `Alt+Left Arrow` or `Alt+Right Arrow` to switch between the different text-mode terminals.

4.5.8 Special care for specific packages

In most cases, packages should upgrade smoothly between lenny and squeeze. There are a small number of cases where some intervention may be required, either before or during the upgrade; these are detailed below on a per-package basis.

4.5.8.1 Evolution

Evolution (the GNOME Desktop mail client) has been updated from version 2.22 to 2.30. This changes the storage format used by the package for local data and there is a possibility of data loss if the upgrade is performed whilst `evolution` is running. Exiting the application itself may not be sufficient, as various related components will continue to run in the background. To avoid any potential issues, it is recommended that you completely exit your desktop environment before beginning the upgrade to squeeze.

As part of the upgrade process, `evolution` will check whether any related processes are running and will recommend that they be closed. A secondary check for processes will then be performed; if necessary, a choice will be offered between allowing the remaining processes to be killed or aborting the upgrade in order to resolve the situation by hand.

4.6 Upgrading your kernel and related packages

This section explains how to upgrade your kernel and identifies potential issues related to this upgrade. You can either install one of the `linux-image-*` packages provided by Debian, or compile a customized kernel from source.

Note that a lot of information in this section is based on the assumption that you will be using one of the modular Debian kernels, together with `initramfs-tools` and `udev`. If you choose to use a custom kernel that does not require an `initrd` or if you use a different `initrd` generator, some of the information may not be relevant for you.

4.6.1 Installing the kernel metapackage

When you `dist-upgrade` from `lenny` to `squeeze`, it is strongly recommended that you install a new `linux-image-2.6-*` metapackage. This package may be installed automatically by the `dist-upgrade` process. You can verify this by running:

```
# dpkg -l "linux-image*" | grep ^ii
```

If you do not see any output, then you will need to install a new `linux-image` package by hand. To see a list of available `linux-image-2.6` metapackages, run:

```
# apt-cache search linux-image-2.6- | grep -v transition
```

If you are unsure about which package to select, run `uname -r` and look for a package with a similar name. For example, if you see `'2.6.26-2-686'`, it is recommended that you install `linux-image-2.6-686`. You may also use `apt-cache` to see a long description of each package in order to help choose the best one available. For example:

```
# apt-cache show linux-image-2.6-686
```

You should then use `apt-get install` to install it. Once this new kernel is installed you should reboot at the next available opportunity to get the benefits provided by the new kernel version.

For the more adventurous there is an easy way to compile your own custom kernel on Debian GNU/Linux. Install the `kernel-package` tool and read the documentation in `/usr/share/doc/kernel-package`. Alternatively, you can also use the kernel sources, provided in the `linux-source-2.6` package. You can make use of the `deb-pkg` target available in the sources' makefile for building a binary package. There are some differences in these two approaches, please consult the respective package's documentation.

If possible, it is to your advantage to upgrade the kernel package separately from the main `dist-upgrade` to reduce the chances of a temporarily non-bootable system. Note that this should only be done after the minimal upgrade process described in Section 4.4.4.

4.6.2 Device enumeration reordering

In `lenny` and later, a new kernel mechanism for hardware discovery may change the order in which devices are discovered on your system on each boot, affecting the device names assigned to them. For example, if you have two network adapters that are associated with two different drivers, the devices `eth0` and `eth1` refer to may be swapped.

For network devices, this reordering is normally avoided by the definitions at `/etc/udev/rules.d/70-persistent-net.rules` for `udev`. Since these rules were already in place in `lenny`, no additional action should be required when upgrading to `squeeze` to get the benefit of stable network device names. Please note, however, that this `udev` mechanism means that a given network device name is tied to a particular piece of hardware; if you, for instance, exchange ethernet adapters in a deployed `squeeze` system, the new adapter will get a new interface name instead of using the existing one. To reuse an existing device name for new hardware, you will need to delete the associated entry from `/etc/udev/rules.d/70-persistent-net.rules`.

For storage devices, you may be able to avoid this reordering by using `initramfs-tools` and configuring it to load storage device driver modules in the same order they are currently loaded. However, in light of other changes to the storage subsystem of the Linux kernel as described at Section 5.1.1, this is usually not worth the effort and it is recommended instead to use device names that are guaranteed to

be stable over time, such as the UUID aliases ⁷ in the `/dev/disk/by-uuid/` directory or LVM device names in `/dev/mapper/`.

4.6.3 Boot timing issues (waiting for root device)

If an `initrd` created with `initramfs-tools` is used to boot the system, in some cases the creation of device files by `udev` can happen too late for the boot scripts to act on.

The usual symptoms are that the boot will fail because the root file system cannot be mounted and you are dropped into a debug shell:

```
Gave up waiting for root device. Common problems:
- Boot args (cat /proc/cmdline)
  - Check rootdelay= (did the system wait long enough?)
  - Check root= (did the system wait for the right device?)
- Missing modules (cat /proc/modules; ls /dev)
ALERT! /dev/something does not exist. Dropping to a shell!
(initramfs)
```

But if you check afterwards, all devices that are needed are present in `/dev`. This has been observed in cases where the root file system is on a USB disk or on RAID, especially if LILO is used.

A workaround for this issue is to use the boot parameter `rootdelay=9`. The value for the timeout (in seconds) may need to be adjusted.

4.7 Preparing for the next release

After the upgrade there are several things you can do to prepare for the next release.

- Remove obsolete and unused packages as described in Section 4.9. You should review which configuration files they use and consider purging the packages to remove their configuration files.

4.8 Deprecated components

With the next release of Debian GNU/Linux 7.0 (codenamed wheezy) some features will be deprecated. Users will need to migrate to other alternatives to prevent trouble when updating to 7.0.

This includes the following features:

- OpenVZ and Linux-Vserver: Debian GNU/Linux 6.0 will be the last release to include Linux kernel virtualization featuresets outside of mainline. This means that the OpenVZ and Linux-Vserver featuresets should be considered deprecated, and users should migrate to linux-2.6 upstream merged virtualization solutions like KVM, Linux Containers or Xen.
- The `gdm` package (GNOME Display Manager version 2.20) will be obsoleted by `gdm3`, a rewritten version. See Section 5.6 for more information.

4.9 Obsolete packages

Introducing several thousand new packages, squeeze also retires and omits more than four thousand old packages that were in lenny. It provides no upgrade path for these obsolete packages. While nothing prevents you from continuing to use an obsolete package where desired, the Debian project will usually discontinue security support for it a year after squeeze's release⁸, and will not normally provide other support in the meantime. Replacing them with available alternatives, if any, is recommended.

There are many reasons why packages might have been removed from the distribution: they are no longer maintained upstream; there is no longer a Debian Developer interested in maintaining the packages; the functionality they provide has been superseded by different software (or a new version);

⁷ Some devices, such as those used by crypt, RAID or LVM have stable non-UUID identifiers. In these cases you should use the name of the devices, which are already unambiguous and stable.

⁸ Or for as long as there is not another release in that time frame. Typically only two stable releases are supported at any given time.

or they are no longer considered suitable for squeeze due to bugs in them. In the latter case, packages might still be present in the “unstable” distribution.

Detecting which packages in an updated system are “obsolete” is easy since the package management front-ends will mark them as such. If you are using **aptitude**, you will see a listing of these packages in the “Obsolete and Locally Created Packages” entry. **dselect** provides a similar section but the listing it presents might differ.

Also, if you have used **aptitude** or **apt-get** to manually install packages in lenny it will have kept track of those packages you manually installed and will be able to mark as obsolete those packages pulled in by dependencies alone which are no longer needed if a package has been removed. **aptitude** and **apt**, unlike **deborphan**, will not mark for removal packages that you manually installed, as opposed to those that were automatically installed through dependencies. To remove automatically installed packages that are no longer used, run:

```
# apt-get autoremove
```

There are additional tools you can use to find obsolete packages such as **deborphan**, **debfoister** or **crufft**. **deborphan** is highly recommended, although it will (in default mode) only report obsolete libraries: packages in the “libs” or “oldlibs” sections that are not used by any other packages. Do not blindly remove the packages these tools present, especially if you are using aggressive non-default options that are prone to produce false positives. It is highly recommended that you manually review the packages suggested for removal (i.e. their contents, size and description) before you remove them.

The **Debian Bug Tracking System** (<http://bugs.debian.org/>) often provides additional information on why the package was removed. You should review both the archived bug reports for the package itself and the archived bug reports for the **ftp.debian.org pseudo-package** (<http://bugs.debian.org/cgi-bin/pkgreport.cgi?pkg=ftp.debian.org&archive=yes>).

The list of obsolete packages includes:

- The plone content management suite. This has been done on request by the developers to use the Unified Installer for Linux, which they consider their only supported deployment platform. The recommended tool for installing Plone on a Debian GNU/Linux system is the Unified Installer, available for download from <http://plone.org/>
- **nessus**, the vulnerability scanning server and its associated libraries and other software. It has been deprecated in favor of the software provided by OpenVAS which includes **openvas-server** and **openvas-client**. As there is no automatic upgrade path you will have to install OpenVAS and manually move over your Nessus service configuration (users, certificates, etc.) to OpenVAS.
- **postgresql-8.3**, successor is **postgresql-8.4**.
- **mysql-server-5.0**, successor is **mysql-server-5.1**.
- **python2.4**, successor is **python2.6**.
- Java 5 software including the packages **sun-java5-jre** and **sun-java5-bin**, successor is Java 6: **sun-java6-jre** and associated packages.
- **apt-proxy** is no longer provided, alternatives to this tool include **apt-cacher-ng**, **apt-cacher** and **approx**. Although no automatic upgrade path exists, user of **apt-proxy** can switch to these alternatives by manually installing any of these packages.
- Some of Xorg's video drivers are no longer available in squeeze and are obsolete. This includes **xserver-xorg-video-cyrix**, **xserver-xorg-video-i810**, **xserver-xorg-video-imstt**, **xserver-xorg-video-nsc**, **xserver-xorg-video-sunbw2**, and **xserver-xorg-video-vga**. They might be removed through the upgrade. Users should install **xserver-xorg-video-all** instead.
- The utility used in lenny to display a splash image at boot time, **usplash**, is no longer available. It has been replaced by **plymouth**.

4.9.1 Dummy packages

Some packages from lenny have been split into several packages in squeeze, often to improve system maintainability. To ease the upgrade path in such cases, squeeze often provides “dummy” packages: empty packages that have the same name as the old package in lenny with dependencies that cause the new packages to be installed. These “dummy” packages are considered obsolete packages after the upgrade and can be safely removed.

Most (but not all) dummy packages' descriptions indicate their purpose. Package descriptions for dummy packages are not uniform, however, so you might also find **deborphan** with the `--guess-*` options (e.g. `--guess-dummy`) useful to detect them in your system. Note that some dummy packages are not intended to be removed after an upgrade but are, instead, used to keep track of the current available version of a program over time.

Chapter 5

Issues to be aware of for squeeze

5.1 Potential problems

Sometimes, changes introduced in a new release have side-effects we cannot reasonably avoid, or they expose bugs somewhere else. This section documents issues we are aware of. Please also read the errata, the relevant packages' documentation, bug reports and other information mentioned in Section 6.1.

5.1.1 Migration of disk drivers from IDE to PATA subsystem

The new Linux kernel version provides different drivers for some PATA (IDE) controllers. The names of some hard disk, CD-ROM, and tape devices may change.

It is now recommended to identify disk devices in configuration files by label or UUID (unique identifier) rather than by device name, which will work with both old and new kernel versions. Upon upgrading to the squeeze version of the Debian kernel packages, the `linux-base` package will offer to do this conversion for you in the config files for most of the filesystem-related packages on the system, including the various bootloaders included in Debian. If you choose not to update the system configuration automatically, or if you are not using the Debian kernel packages, you must update device IDs yourself before the next system reboot to ensure the system remains bootable.

5.1.2 mdadm metadata format change requires recent Grub

The following only applies to users who want to let the `grub-pc` bootloader load the kernel directly off a RAID device created with `mdadm 3.x` and default values, or when the metadata version is explicitly set using `-e`. Specifically, this includes all arrays created during or after the installation of Debian squeeze. Arrays created with older `mdadm` versions, and RAIDs created with the command-line option `-e 0.9` are not affected.

Versions of `grub-pc` older than 1.98+20100720-1 will not be able to boot directly off a RAID with the 1.x metadata formats (the new default is 1.2). To ensure a bootable system, please make sure to use `grub-pc 1.98+20100720-1` or later, which is provided by Debian squeeze. An unbootable system may be rescued with [Super Grub2 Disk](http://www.supergrubdisk.org/super-grub2-disk/) or [grml](http://grml.org).

5.1.3 pam_userdb.so breakage with newer libdb

Some Berkeley Database version 7 files created with `libdb3` cannot be read by newer `libdb` versions (see bug [#521860](http://bugs.debian.org/521860)). As a workaround, the files can be recreated with `db4.8_load`, from the `db4.8-util` package.

5.1.4 Potential issues with diversions of /bin/sh

If you have previously added a local diversion for `/bin/sh`, or modified the `/bin/sh` symlink to point to somewhere other than `/bin/bash`, then you may encounter problems when upgrading the `dash` or `bash` packages. Note that this includes changes made by allowing other packages (for example `mksh`) to become the default system shell by taking over `/bin/sh`.

If you encounter any such issues, please remove the local diversion and ensure that the symlinks for both `/bin/sh` and its manual page point to the files provided by the `bash` package and then `dpkg-reconfigure --force dash`.

```
dpkg-divert --remove /bin/sh
dpkg-divert --remove /usr/share/man/man1/sh.1.gz

ln -sf bash /bin/sh
ln -sf bash.1.gz /usr/share/man/man1/sh.1.gz
```

5.1.5 Change in kernel policy regarding resource conflicts

The default setting for the `acpi_enforce_resources` parameter in the Linux kernel has changed to be `strict` by default. This can lead some legacy sensor drivers to be denied access to the sensors' hardware. One workaround is to append `acpi_enforce_resources=lax` to the kernel command line.

5.2 LDAP support

A feature in the cryptography libraries used in the LDAP libraries causes programs that use LDAP and attempt to change their effective privileges to fail when connecting to an LDAP server using TLS or SSL. This can cause problems for `suid` programs on systems using `libnss-ldap` like `sudo`, `su` or `schroot` and for `suid` programs that perform LDAP searches like `sudo-ldap`.

It is recommended to replace the `libnss-ldap` package with `libnss-ldapd`, a newer library which uses separate daemon (`nslcd`) for all LDAP lookups. The replacement for `libpam-ldap` is `libpam-ldapd`.

Note that `libnss-ldapd` recommends the NSS caching daemon (`nscd`) which you should evaluate for suitability in your environment before installing. As an alternative to `nscd` you can consider `unscd`.

Further information is available in bugs [#566351](http://bugs.debian.org/566351) (<http://bugs.debian.org/566351>) and [#545414](http://bugs.debian.org/545414) (<http://bugs.debian.org/545414>).

5.3 sieve service moving to its IANA-allocated port

The IANA port allocated for ManageSieve is `4190/tcp`, and the old port used by `timsieved` and other managesieve software in many distributions (`2000/tcp`) is allocated for Cisco SCCP usage, according to [the IANA registry](http://www.iana.org/assignments/port-numbers) (<http://www.iana.org/assignments/port-numbers>).

Starting with the version 4.38 of the Debian `netbase` package, the `sieve` service will be moved from port 2000 to port 4190 in the `/etc/services` file.

Any installs which used the `sieve` service name instead of a numeric port number will switch to the new port number as soon as the services are restarted or reloaded, and in some cases, immediately after `/etc/services` is updated.

This will affect Cyrus IMAP. This may also affect other sieve-enabled software such as DoveCot.

In order to avoid downtime problems, mail cluster administrators using Debian are urged to verify their Cyrus (and probably also DoveCot) installs, and take measures to avoid services moving from port `2000/tcp` to port `4190/tcp` by surprise in either servers or clients.

It is worth noting that:

- `/etc/services` will only be automatically updated if you never made any modifications to it. Otherwise, you will be presented with a prompt by `dpkg` asking you about the changes.
- You can edit `/etc/services` and change the `sieve` port back to 2000 if you want (this is not recommended, though).
- You can edit `/etc/cyrus.conf` and any other relevant configuration files for your mail/webmail cluster (e.g. on the sieve web frontends) ahead of time to force them all to a static port number.
- You can configure cyrus master to listen on both ports (2000 and 4190) at the same time, and thus avoid the problem entirely. This also allows for a much more smooth migration from port 2000 to port 4190.

5.4 Security status of web browsers

Debian 6.0 includes several browser engines which are affected by a steady stream of security vulnerabilities. The high rate of vulnerabilities and partial lack of upstream support in the form of long term branches make it very difficult to support these browsers with backported security fixes. Additionally, library interdependencies make it impossible to update to newer upstream releases. As such, browsers built upon the qtwebkit and khtml engines are included in Squeeze, but not covered by full security support. We will make an effort to track down and backport security fixes, but in general these browsers should not be used against untrusted websites.

For general web browser use we recommend browsers building on the Mozilla xulrunner engine (Iceweasel and Iceape), browsers based on the Webkit engine (e.g. Epiphany) or Chromium. Xulrunner has had a history of good backportability for older releases over the previous release cycles.

Chromium—while built upon the Webkit codebase—is a leaf package, i.e. if backporting becomes no longer feasible, there's still the possibility of upgrading to a later upstream release (which is not possible for the webkit library itself).

Webkit is supported by upstream with a long term maintenance branch.

5.5 KDE desktop

Squeeze is the first Debian release to ship with the full support for the next generation KDE that is based on Qt 4. Most official KDE applications are at version 4.4.5 with the exception of `kdepim` that is at version 4.4.7. You can read the [announcements from the KDE Project](http://www.kde.org/announcements/) (<http://www.kde.org/announcements/>) to learn more about the changes.

5.5.1 Upgrading from KDE 3

KDE 3 Desktop Environment is no longer supported in Debian 6.0. It will be automatically replaced by the new 4.4 series on upgrade. As this is a major change, users should take some precautions in order to ensure as smooth of an upgrade process as possible.

Important



It is discouraged to upgrade while there is an active KDE 3 session on the system. Otherwise, the process might render the running session dysfunctional with the possibility of data loss.

Upon the first login on the upgraded system, existing users will be prompted with the Debian-KDE guided migration procedure called `kaboom` which will assist in the process of migrating the user's personal data and optionally backing up old KDE configuration. For more information, visit [the Kaboom homepage](http://pkg-kde.alioth.debian.org/kaboom.html) (<http://pkg-kde.alioth.debian.org/kaboom.html>).

While KDE 3 based desktop environment is no longer supported, users can still install and use some individual KDE 3 applications since the core libraries and binaries of KDE 3 (`kdelibs`) and Qt 3 are still available in Debian 6.0. However, please note that these applications might not be well integrated with the new environment. What's more, neither KDE 3 nor Qt 3 will be supported in any form in the next Debian release so if you are using them, you are strongly advised to port your software to the new platform.

5.5.2 New KDE metapackages

As noted earlier, Debian 6.0 introduces a new set of KDE related metapackages:

- You are strongly advised to install the `kde-standard` package for normal desktop usage. `kde-standard` will pull in the [KDE Plasma Desktop](http://www.kde.org/workspaces/plasmadesktop/) (<http://www.kde.org/workspaces/plasmadesktop/>) by default, and a selected set of commonly used applications.

- If you want a minimal desktop you can install the `kde-plasma-desktop` package and manually pick the applications you need. This is a rough equivalent of the `kde-minimal` package as shipped in Debian 5.0.
- For small form factor devices, there is an alternative environment called **KDE Plasma Netbook** (<http://www.kde.org/workspaces/plasmanetbook/>) that can be installed with the `kde-plasma-netbook` package. Plasma Netbook and Plasma Desktop can live in the same system and the default can be configured in System Settings (replacement of the former KControl).
- If you want a full set of official KDE applications, you have the possibility to install the `kde-full` package. It will install KDE Plasma Desktop by default.

5.6 GNOME desktop changes and support

There have been many changes in the GNOME desktop environment from the version shipped in lenny to the version in squeeze, you can find more information in the **GNOME 2.30 Release Notes** (<http://library.gnome.org/misc/release-notes/2.30/>). Specific issues are listed below.

5.6.1 GDM 2.20 and 2.30

The GNOME Display Manager (GDM), is kept at version 2.20 for systems upgraded from lenny. This version will still be maintained for the squeeze cycle but it is the last release to do so. Newly installed systems will get GDM 2.30 instead, provided by the `gdm3` package. Because of incompatibilities between both versions, this upgrade is not automatic, but it is recommended to install `gdm3` after the upgrade to squeeze. This should be done from the console, or with only one open GNOME session. Note that settings from GDM 2.20 will **not** be migrated. For a standard desktop system, however, simply installing `gdm3` should be enough.

5.6.2 Device and other administrative permissions

Specific permissions on devices are granted automatically to the user currently logged on physically to the system: video and audio devices, network roaming, power management, device mounting. The `cdrom`, `floppy`, `audio`, `video`, `plugdev` and `powerdev` groups are no longer useful. See the `consolekit` documentation for more information.

Most graphical programs requiring root permissions now rely on **PolicyKit** (<http://www.freedesktop.org/wiki/Software/PolicyKit>) to do so, instead of `gksu`. The recommended way to give a user administrative rights is to add it to the `sudo` group.

5.6.3 network-manager and ifupdown interaction

Upon upgrading the `network-manager` package, interfaces configured in `/etc/network/interfaces` to use DHCP with no other options will be disabled in that file, and handled by NetworkManager instead. Therefore the `ifup` and `ifdown` commands will not work. These interfaces can be managed using the NetworkManager frontends instead, see **the NetworkManager documentation** (<http://live.gnome.org/NetworkManager/SystemSettings>).

Conversely, any interfaces configured in `/etc/network/interfaces` with more options will be ignored by NetworkManager. This applies in particular to wireless interfaces used during the installation of Debian (see bug **#606268** (<http://bugs.debian.org/606268>)).

5.7 Graphics stack changes

There have been a number of changes to the X stack in Debian 6.0. This section lists the most important and user-visible.

5.7.1 Obsolete Xorg drivers

The `cyrix`, `imstt`, `sunbw2` and `vga` Xorg video drivers are no longer provided. Users should switch to a generic such as `vesa` or `fbdev` instead.

The old `via` driver was no longer maintained, and has been replaced by the `openchrome` driver, which will be used automatically after the upgrade.

The `nv` and `radeonhd` drivers are still present in this release, but are deprecated. Users should consider the `nouveau` and `radeon` drivers instead, respectively.

The `calcomp`, `citron`, `digitaledge`, `dmc`, `dynapro`, `elo2300`, `fpit`, `hyperpen`, `jamstudio`, `magellan`, `microtouch`, `mutouch`, `palmax`, `spaceorb`, `summa`, `tek4957` and `ur98` X input drivers have been discontinued and are not included in this release. Users of these devices might want to switch to a suitable kernel driver and the `evdev` X driver. For many serial devices, the `inputattach` utility allows attaching them to a Linux input device which can be recognized by the `evdev` X driver.

5.7.2 Kernel mode setting

Kernel drivers for Intel (starting from `i830`), ATI/AMD (from the original Radeon to the Radeon HD 5xxx “Evergreen” series) and for NVIDIA graphics chipsets now support native mode setting.

5.7.3 Input device hotplug

The Xorg X server included in Debian 6.0 provides improved support for hotplugging of input devices (mice, keyboards, tablets, ...). The old `xserver-xorg-input-kbd` and `xserver-xorg-input-mouse` packages are replaced by `xserver-xorg-input-evdev`, which requires a kernel with the `CONFIG_INPUT_EVDEV` option enabled. Additionally, some of the keycodes produced by this driver differ from those traditionally associated with the same keys. Users of programs like `xmodmap` and `xbindkeys` will need to adjust their configurations for the new keycodes.

5.7.4 X server “zapping”

Traditionally, the `Ctrl-Alt-Backspace` combination would kill the X server. This combination is no longer active by default, but can be re-enabled by reconfiguring the `keyboard-configuration` package (system-wide), or using your desktop environment’s keyboard preferences application.

5.8 Munin web path change

For `squeeze`, the default location for the generated web content of `munin` has been changed from `/var/www/munin` to `/var/cache/munin/www` and therefore `/etc/munin/munin.conf` needs to be adapted on upgrades, if it has been changed by the admin. If you are upgrading, please read `/usr/share/doc/munin/NEWS.Debian.gz`.

5.9 Shorewall upgrade instructions

Users of the `shorewall` firewall should read the instructions at <http://www.shorewall.net/LennyToSqueeze.html>, also available as `/usr/share/doc/shorewall-doc/html/LennyToSqueeze.html` in the `shorewall-doc` package, upon upgrading to Debian 6.0.

Chapter 6

More information on Debian GNU/Linux

6.1 Further reading

Beyond these release notes and the installation guide, further documentation on Debian GNU/Linux is available from the Debian Documentation Project (DDP), whose goal is to create high-quality documentation for Debian users and developers. Documentation, including the Debian Reference, Debian New Maintainers Guide, and Debian FAQ are available, and many more. For full details of the existing resources see the [Debian Documentation website](http://www.debian.org/doc/) (<http://www.debian.org/doc/>) and the [Debian Wiki website](http://wiki.debian.org/) (<http://wiki.debian.org/>)

Documentation for individual packages is installed into `/usr/share/doc/package`. This may include copyright information, Debian specific details and any upstream documentation.

6.2 Getting help

There are many sources of help, advice and support for Debian users, but these should only be considered if research into documentation of the issue has exhausted all sources. This section provides a short introduction into these which may be helpful for new Debian users.

6.2.1 Mailing lists

The mailing lists of most interest to Debian users are the `debian-user` list (English) and other `debian-user-language` lists (for other languages). For information on these lists and details of how to subscribe see <http://lists.debian.org/>. Please check the archives for answers to your question prior to posting and also adhere to standard list etiquette.

6.2.2 Internet Relay Chat

Debian has an IRC channel dedicated to the support and aid of Debian users located on the OFTC IRC network. To access the channel, point your favorite IRC client at `irc.debian.org` and join `#debian`.

Please follow the channel guidelines, respecting other users fully. The guidelines are available at the [Debian Wiki](http://wiki.debian.org/DebianIRC) (<http://wiki.debian.org/DebianIRC>).

For more information on OFTC please visit the [website](http://www.oftc.net/) (<http://www.oftc.net/>).

6.3 Reporting bugs

We strive to make Debian GNU/Linux a high quality operating system, however that does not mean that the packages we provide are totally free of bugs. Consistent with Debian's "open development" philosophy and as a service to our users, we provide all the information on reported bugs at our own Bug Tracking System (BTS). The BTS is browseable at <http://bugs.debian.org/>.

If you find a bug in the distribution or in packaged software that is part of it, please report it so that it can be properly fixed for future releases. Reporting bugs requires a valid email address. We ask

for this so that we can trace bugs and developers can get in contact with submitters should additional information be needed.

You can submit a bug report using the program **reportbug** or manually using email. You can read more about the Bug Tracking System and how to use it by reading the reference documentation (available at `/usr/share/doc/debian` if you have `doc-debian` installed) or online at the **Bug Tracking System** (<http://bugs.debian.org/>).

6.4 Contributing to Debian

You do not need to be an expert to contribute to Debian. By assisting users with problems on the various user support **lists** (<http://lists.debian.org/>) you are contributing to the community. Identifying (and also solving) problems related to the development of the distribution by participating on the development **lists** (<http://lists.debian.org/>) is also extremely helpful. To maintain Debian's high quality distribution, **submit bugs** (<http://bugs.debian.org/>) and help developers track them down and fix them. If you have a way with words then you may want to contribute more actively by helping to write **documentation** (<http://www.debian.org/doc/cvs>) or **translate** (<http://www.debian.org/international/>) existing documentation into your own language.

If you can dedicate more time, you could manage a piece of the Free Software collection within Debian. Especially helpful is if people adopt or maintain items that people have requested for inclusion within Debian. The **Work Needing and Prospective Packages database** (<http://www.debian.org/devel/wnpp/>) details this information. If you have an interest in specific groups then you may find enjoyment in contributing to some of Debian's subprojects which include ports to particular architectures, **Debian Jr.** (<http://www.debian.org/devel/debian-jr/>) and **Debian Med** (<http://www.debian.org/devel/debian-med/>).

In any case, if you are working in the free software community in any way, as a user, programmer, writer or translator you are already helping the free software effort. Contributing is rewarding and fun, and as well as allowing you to meet new people it gives you that warm fuzzy feeling inside.

Chapter 7

Glossary

ACPI

Advanced Configuration and Power Interface

ALSA

Advanced Linux Sound Architecture

APM

Advanced Power Management

BD

Blu-ray Disc

CD

Compact Disc

CD-ROM

Compact Disc Read Only Memory

DHCP

Dynamic Host Configuration Protocol

DNS

Domain Name System

DVD

Digital Versatile Disc

GIMP

GNU Image Manipulation Program

GNU

GNU's Not Unix

GPG

GNU Privacy Guard

IDE

Integrated Drive Electronics

LDAP

Lightweight Directory Access Protocol

LILO

LIinux LOader

LSB

Linux Standard Base

LVM

Logical Volume Manager

MTA

Mail Transport Agent

NFS

Network File System

NIC

Network Interface Card

NIS

Network Information Service

OSS

Open Sound System

RAID

Redundant Array of Independent Disks

RPC

Remote Procedure Call

SATA

Serial Advanced Technology Attachment

SSL

Secure Sockets Layer

TLS

Transport Layer Security

USB

Universal Serial Bus

UUID

Universally Unique Identifier

VGA

Video Graphics Array

WPA

Wi-Fi Protected Access

Appendix A

Managing your lenny system before the upgrade

This appendix contains information on how to make sure you can install or upgrade lenny packages before you upgrade to squeeze. This should only be necessary in specific situations.

A.1 Upgrading your lenny system

Basically this is no different than any other upgrade of lenny you've been doing. The only difference is that you first need to make sure your package list still contains references to lenny as explained in Section A.2.

If you upgrade your system using a Debian mirror, it will automatically be upgraded to the latest lenny point release.

A.2 Checking your sources list

If any of the lines in your `/etc/apt/sources.list` refer to 'stable', you are effectively already “using” squeeze. This might not be what you want if you are not ready yet for the upgrade. If you have already run `apt-get update`, you can still get back without problems following the procedure below.

If you have also already installed packages from squeeze, there probably is not much point in installing packages from lenny anymore. In that case you will have to decide for yourself whether you want to continue or not. It is possible to downgrade packages, but that is not covered here.

Open the file `/etc/apt/sources.list` with your favorite editor (as `root`) and check all lines beginning with `deb http:` or `deb ftp:` for a reference to “stable”. If you find any, change `stable` to `lenny`.

If you have any lines starting with `deb file:`, you will have to check for yourself if the location they refer to contains a lenny or a squeeze archive.

Important



Do not change any lines that begin with `deb cdrom:`. Doing so would invalidate the line and you would have to run **apt-cdrom** again. Do not be alarmed if a 'cdrom' source line refers to “unstable”. Although confusing, this is normal.

If you've made any changes, save the file and execute

```
# apt-get update
```

to refresh the package list.

A.3 Upgrade legacy locales to UTF-8

If your system is localised and is using a locale that is not based on UTF-8 you should strongly consider switching your system over to using UTF-8 locales. In the past, there have been bugs identified that manifest itself only when using a non-UTF-8 locale. On the desktop, such legacy locales are supported through ugly hacks in the libraries internals, and we cannot decently provide support for users who still use them.

To configure your system's locale you can run **dpkg-reconfigure locales**. Ensure you select an UTF-8 locale when you are presented with the question asking which locale to use as a default in the system. In addition, you should review the locale settings of your users and ensure that they do not have legacy locales definitions in their configuration environment.

Appendix B

Contributors to the Release Notes

Many people helped with the release notes, including, but not limited to

Adam Di Carlo, Andreas Barth, Andrei Popescu, Anne Bezemer, Bob Hilliard, Charles Plessy, Christian Perrier, Daniel Baumann, Eddy Petrișor, Emmanuel Kasper, Esko Arajärvi, Frans Pop, Giovanni Ragnani, Gordon Farquharson, Javier Fernández-Sanguino Peña, Jens Seidel, Jonas Meurer, Josip Rodin, Justin B Rye, LaMont Jones, Luk Claes, Martin Michlmayr, Michael Biebl, Moritz Mühlenhoff, Noah Meyerhans, Noritada Kobayashi, Osamu Aoki, Peter Green, Rob Bradford, Samuel Thibault, Simon Bienlein, Simon Paillard, Stefan Fritsch, Steve Langasek, Steve McIntyre, Tobias Scherer, Vincent McIntyre, and W. Martin Borgert.

This document has been translated into many languages. Many thanks to the translators!

Index

A

Abiword, 4
Apache, 4

B

BIND, 4
Blu-ray, 4

C

CD, 4
Cherokee, 4
Courier, 4

D

Debian Med, 6
Debian Science, 6
Dia, 4
DocBook XML, 2
DVD, 4

E

Ekiga, 4
Evolution, 4
Exim, 4

F

Firefox, 4

G

GCC, 4
GIMP, 4
GNOME, 4
GNUCash, 4
GNUmeric, 4

I

isohybrid, 4

K

KDE, 4
KOffice, 4

L

LDAP, 6, 26
LILO, 21
Linux Standard Base, 4
LXDE, 4

M

Mozilla, 4
MySQL, 4

N

NeuroDebian, 6
neuroimaging research, 6

O

OpenOffice.org, 4
OpenSSH, 4

P

packages
ants, 6
approx, 22
apt, 1, 2, 13, 22
apt-cacher, 22
apt-cacher-ng, 22
apt-proxy, 22
aptitude, 5, 15
autofs, 18
autofs5, 18
bash, 25, 26
console-setup, 5
consolekit, 28
dash, 25
db4.8-util, 25
dblatex, 2
debian-goodies, 15
dhcp3, 18
discover, 7
doc-debian, 32
docbook-xsl, 2
evolution, 19
firmware-linux, 4
gdm, 21
gdm3, 21, 28
gksu, 28
grub-pc, 25
icedove, 4
iceweasel, 4
initramfs-tools, 10, 20, 21
insserv, 5
isc-dhcp, 18
kaboom, 27
kde-full, 28
kde-minimal, 28
kde-plasma-desktop, 28
kde-plasma-netbook, 28
kde-standard, 27
kdelibs, 27
kdepim, 27
kernel-package, 20
keyboard-configuration, 29
libnss-ldap, 6, 26
libnss-ldapd, 6, 26
libpam-ldap, 6, 26
libpam-ldapd, 6, 26
linux-base, 25
linux-image-*, 20
linux-image-2.6-686, 20
linux-source-2.6, 20
localepurge, 15
madwifi-source, 18
mdadm, 25
mksh, 25
mrtrix, 6

mysql-server-5.0, 22
mysql-server-5.1, 22
nessus, 22
netbase, 26
network-manager, 28
nipy, 6
nscd, 26
odin, 6
openvas-client, 22
openvas-server, 22
plone, 22
plymouth, 22
popularity-contest, 15
postgresql-8.3, 22
postgresql-8.4, 22
psychopy, 6
python2.4, 18, 22
python2.6, 18, 22
release-notes, 1
shorewall, 29
shorewall-doc, 29
splashy, 11
sudo-ldap, 26
sun-java5-bin, 22
sun-java5-jre, 22
sun-java6-jre, 22
tinc, 11
udev, 17, 20, 21
unscd, 26
upgrade-reports, 1
usplash, 22
xmlroff, 2
xserver-xorg-input-evdev, 29
xserver-xorg-input-kbd, 29
xserver-xorg-input-mouse, 29
xserver-xorg-video-all, 22
xserver-xorg-video-cyrix, 22
xserver-xorg-video-i810, 22
xserver-xorg-video-imstt, 22
xserver-xorg-video-nsc, 22
xserver-xorg-video-sunbw2, 22
xserver-xorg-video-vga, 22
xsltproc, 2

PHP, 4
Pidgin, 4
Postfix, 4
PostgreSQL, 4

T
Thunderbird, 4
Tomcat, 4

X
Xfce, 4