

Release Notes for Debian 9 (stretch), 32-bit MIPS (big endian)

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

June 25, 2019

Release Notes for Debian 9 (stretch), 32-bit MIPS (big endian)

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.

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 9	3
2.1	Supported architectures	3
2.2	What's new in the distribution?	3
2.2.1	CDs, DVDs, and BDs	4
2.2.2	Security	4
2.2.3	GCC versions	4
2.2.4	MariaDB replaces MySQL	4
2.2.5	Improvements to APT and archive layouts	5
2.2.6	New deb.debian.org mirror	5
2.2.7	Move to "Modern" GnuPG	5
2.2.8	A new archive for debug symbols	6
2.2.9	New method for naming network interfaces	6
2.2.10	News from Debian Med Blend	6
2.2.11	The Xorg server no longer requires root	6
3	Installation System	9
3.1	What's new in the installation system?	9
3.1.1	Major changes	9
3.1.2	Automated installation	10
4	Upgrades from Debian 8 (jessie)	11
4.1	Preparing for the upgrade	11
4.1.1	Back up any data or configuration information	11
4.1.2	Inform users in advance	11
4.1.3	Prepare for downtime on services	11
4.1.4	Prepare for recovery	12
4.1.4.1	Debug shell during boot using initrd	12
4.1.4.2	Debug shell during boot using systemd	12
4.1.5	Prepare a safe environment for the upgrade	13
4.2	Checking system status	13
4.2.1	Review actions pending in package manager	13
4.2.2	Disabling APT pinning	14
4.2.3	Checking packages status	14
4.2.4	The proposed-updates section	14
4.2.5	Unofficial sources	15
4.3	Preparing sources for APT	15
4.3.1	Adding APT Internet sources	15
4.3.2	Adding APT sources for a local mirror	16
4.3.3	Adding APT sources from optical media	16
4.4	Upgrading packages	16
4.4.1	Recording the session	17
4.4.2	Updating the package list	17
4.4.3	Make sure you have sufficient space for the upgrade	17
4.4.4	Minimal system upgrade	19
4.4.5	Upgrading the system	19
4.5	Possible issues during upgrade	20
4.5.1	Dist-upgrade fails with "Could not perform immediate configuration"	20
4.5.2	Expected removals	20
4.5.3	Conflicts or Pre-Depends loops	20

4.5.4	File conflicts	20
4.5.5	Configuration changes	21
4.5.6	Change of session to console	21
4.6	Upgrading your kernel and related packages	21
4.6.1	Installing a kernel metapackage	21
4.7	Preparing for the next release	22
4.7.1	Purging removed packages	22
4.8	Obsolete packages	22
4.8.1	Dummy packages	23
5	Issues to be aware of for stretch	25
5.1	Upgrade specific items for stretch	25
5.1.1	Late mounting of /usr is no longer supported	25
5.1.2	FTP access to Debian hosted mirrors will be removed	25
5.1.3	Noteworthy obsolete packages	25
5.1.4	Things to do post upgrade before rebooting	26
5.1.5	Executables are now compiled as position independent executables (PIE) by default	26
5.1.5.1	Behavior changes of PIE for system administrators and developers	26
5.1.6	Most LSB compatibility packages have been removed	27
5.1.7	32-bit MIPS now requires an R2 processor	27
5.2	Limitations in security support	27
5.2.1	Security status of web browsers	27
5.2.2	Lack of security support for the ecosystem around libv8 and Node.js	28
5.3	Package specific issues	28
5.3.1	Older ciphers and SSH1 protocol disabled in OpenSSH by default	28
5.3.2	Possible backwards incompatible changes to APT	28
5.3.2.1	APT now fetches files as an unprivileged user (<code>_apt</code>)	28
5.3.2.2	New APT pinning engine	28
5.3.2.3	New requirements for APT repository	29
5.3.3	Desktops will migrate to <code>libinput</code> Xorg driver	29
5.3.4	Upstart removed	29
5.3.5	The <code>debhelper</code> tool now generates <code>dbgsym</code> packages by default	30
5.3.6	OpenSSL related changes	30
5.3.7	Perl changes that may break third-party software	30
5.3.8	PostgreSQL PL/Perl incompatibility	31
5.3.9	<code>net-tools</code> will be deprecated in favor of <code>iproute2</code>	31
5.3.10	The <code>_netdev</code> mount option is recommended when using AoE (ATA over ethernet) devices	32
5.3.11	Harmless “Unescaped ... in regex is deprecated, ...” warnings during upgrade	32
5.3.12	SELinux policy store migration	32
5.3.13	iSCSI Enterprise Target no longer supported	32
6	More information on Debian	33
6.1	Further reading	33
6.2	Getting help	33
6.2.1	Mailing lists	33
6.2.2	Internet Relay Chat	33
6.3	Reporting bugs	33
6.4	Contributing to Debian	34
7	Glossary	35
A	Managing your jessie system before the upgrade	37
A.1	Upgrading your jessie system	37
A.2	Checking your sources list	37
A.3	Removing obsolete configuration files	38
A.4	Upgrade legacy locales to UTF-8	38
B	Contributors to the Release Notes	39

Index

41

Chapter 1

Introduction

This document informs users of the Debian distribution about major changes in version 9 (codenamed stretch).

The release notes provide information on how to upgrade safely from release 8 (codenamed jessie) 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 <https://www.debian.org/releases/stretch/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 jessie). If you need to upgrade from older releases, we suggest you read previous editions of the release notes and upgrade to jessie 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](https://bugs.debian.org/) (<https://bugs.debian.org/>) against the `release-notes` package. You might first want to review the [existing bug reports](https://bugs.debian.org/release-notes) (<https://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 jessie to stretch. If you are willing to share information please file a bug in the [bug tracking system](https://bugs.debian.org/) (<https://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** (<https://anonscm.debian.org/viewvc/ddp/manuals/trunk/release-notes/>) to access its files individually through the web and see their changes. For more information on how to access SVN please consult the **Debian Documentation Project SVN information pages** (<https://www.debian.org/doc/cvs>).

Chapter 2

What's new in Debian 9

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

2.1 Supported architectures

Debian 9 introduces one new architecture:

- 64-bit little-endian MIPS (`mips64el`)

Debian 9 regrettably removes support for the following architecture:

- PowerPC (`powerpc`)

The following are the officially supported architectures for Debian 9:

- 32-bit PC (`i386`) and 64-bit PC (`amd64`)
- 64-bit ARM (`arm64`)
- ARM EABI (`armel`)
- ARMv7 (EABI hard-float ABI, `armhf`)
- MIPS (`mips` (big-endian) and `mipsel` (little-endian))
- 64-bit little-endian MIPS (`mips64el`)
- 64-bit little-endian PowerPC (`ppc64el`)
- IBM System z (`s390x`)

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

2.2 What's new in the distribution?

This new release of Debian again comes with a lot more software than its predecessor jessie; the distribution includes over 15346 new packages, for a total of over 51687 packages. Most of the software in the distribution has been updated: over 29859 software packages (this is 57% of all packages in jessie). Also, a significant number of packages (over 6739, 13% of the packages in jessie) 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; see Section 4.8.

Debian again ships with several desktop applications and environments. Among others it now includes the desktop environments GNOME 3.22, KDE Plasma 5.8, LXDE, LXQt 0.11, MATE 1.16, and Xfce 4.12.

Productivity applications have also been upgraded, including the office suites:

- LibreOffice is upgraded to version 5.2;
- Calligra is upgraded to 2.9.

Updates of other desktop applications include the upgrade to Evolution 3.22. Among many others, this release also includes the following software updates:

Package	Version in 8 (jessie)	Version in 9 (stretch)
BIND DNS Server	9.9	9.10
Emacs	24.4	24.5 and 25.1
Exim default e-mail server	4.84	4.88
GNU Compiler Collection as default compiler	4.9	6.3
GnuPG	1.4	2.1
Inkscape	0.48	0.91
the GNU C library	2.19	2.24
Linux kernel image	3.16 series	4.9 series
MariaDB	10.0	10.1
Nginx	1.6	1.10
OpenJDK	7	8
OpenSSH	6.7p1	7.4p1
Perl	5.20	5.24
PHP	5.6	7.0
Postfix MTA	2.11	3.1
PostgreSQL	9.4	9.6
Python 3	3.4	3.5
Samba	4.1	4.5
Vim	7	8

2.2.1 CDs, DVDs, and BDs

The official Debian distribution now ships on 12 to 14 binary DVDs (depending on the architecture) and 12 source DVDs. 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 is also released as Blu-ray (BD) and dual layer Blu-ray (DLBD) images for the `amd64` and `i386` architectures, and also for source code. Debian used to be released as a very large set of CDs for each architecture, but with the stretch release these have been dropped.

2.2.2 Security

For the stretch release, the Debian version of the GNU GCC 6 compiler now defaults to compiling "position independent executables" (PIE). Accordingly the vast majority of all executables will now support **address space layout randomization (ASLR)** (https://en.wikipedia.org/wiki/Address_space_layout_randomization), which is a mitigation for a number of exploits that are now probabilistic rather than deterministic.

2.2.3 GCC versions

Debian stretch includes only version 6 of the GNU GCC compiler, which may impact users expecting version 4.x or 5.x to be available. See the **GCC5** (<https://wiki.debian.org/GCC5>) and **GCC6** (<https://wiki.debian.org/GCC6>) wiki pages for more information about the transition.

2.2.4 MariaDB replaces MySQL

MariaDB is now the default MySQL variant in Debian, at version 10.1. The stretch release introduces a new mechanism for switching the default variant, using metapackages created from the `mysql-defaults` source package. For example, installing the metapackage `default-mysql-server` will install `mariadb-server-10.1`. Users who had `mysql-server-5.5` or `mysql-server-5.6` will have it

removed and replaced by the MariaDB equivalent. Similarly, installing `default-mysql-client` will install `mariadb-client-10.1`.

IMPORTANT



Note that the database binary data file formats are not backwards compatible, so once you have upgraded to MariaDB 10.1 you will not be able to switch back to any previous version of MariaDB or MySQL unless you have a proper database dump. Therefore, before upgrading, please make backups of all important databases with an appropriate tool such as **mysqldump**.

The `virtual-mysql-*` and `default-mysql-*` packages will continue to exist. MySQL continues to be maintained in Debian, in the unstable release. See the [Debian MySQL Team wiki page](https://wiki.debian.org/Teams/MySQL) (<https://wiki.debian.org/Teams/MySQL>) for current information about the mysql-related software available in Debian.

2.2.5 Improvements to APT and archive layouts

The `apt` package manager has seen a number of improvements since `jessie`. Most of these apply to `aptitude` as well. Following are selected highlights of some of these.

On the security side, APT now rejects weaker checksums by default (e.g. SHA1) and attempts to download as an unprivileged user. Please refer to Section 5.3.2.3 and Section 5.3.2.1 for more information.

The APT-based package managers have also gotten a number of improvements that will remove the annoying “hash sum mismatch” warning that occurs when running `apt` during a mirror synchronization. This happens via the new `by-hash` layout, which enables APT to download metadata files by their content hash.

If you use third-party repositories, you may still experience these intermittent issues, if the vendor does not provide the `by-hash` layout. Please recommend them to adopt this layout change. A very short technical description is available in the [Repository format description](https://wiki.debian.org/DebianRepository/Format) (<https://wiki.debian.org/DebianRepository/Format>).

While this may be mostly interesting for mirror administrators, APT in `stretch` can use DNS (SRV) records to locate an HTTP backend. This is useful for providing a simple DNS name and then managing backends via DNS rather than using a “redirector” service. This feature is also used by the new Debian mirror described in Section 2.2.6.

2.2.6 New `deb.debian.org` mirror

Debian now provides a new additional service called deb.debian.org (<https://deb.debian.org>). It provides the content of the main archive, the security archive, ports and even our new debug archive (see Section 2.2.8) under a single easy to remember hostname.

This service relies on the new DNS support in APT, but will fall back to a regular redirect for HTTPS access or older versions of APT. More details are provided on deb.debian.org (<https://deb.debian.org>).

Thanks to Fastly and Amazon CloudFront for sponsoring the CDN backends behind this service.

2.2.7 Move to “Modern” GnuPG

The `stretch` release is the first version of Debian to feature the “modern” branch of GnuPG in the `gnupg` package. This brings with it elliptic curve cryptography, better defaults, a more modular architecture, and improved smartcard support. The modern branch also explicitly does not support some older, known-broken formats (like PGPv3). See `/usr/share/doc/gnupg/README.Debian` for more information.

We will continue to supply the “classic” branch of GnuPG as `gnupg1` for people who need it, but it is now deprecated.

2.2.8 A new archive for debug symbols

NOTE



This section is mostly interesting for developers or if you wish to attach a full stack trace to a crash report.

Previously, the main Debian archive would include packages containing debug symbols for selected libraries or programs. With stretch, most of these have been moved to a separate archive called the `debian-debug` archive. This archive contains the debug symbol packages for the vast majority of all packages provided by Debian.

If you want to fetch such debug packages, please include the following in your APT sources:

```
deb http://debug.mirrors.debian.org/debian-debug/ stretch-debug main
```

Alternatively, you can also fetch them from snapshot.debian.org (<http://snapshot.debian.org>).

Once enabled, you can now fetch debug symbols for the package in question by installing `pkg-dbg`. Please note that individual packages may still provide a `pkg-dbg` package in the main archive instead of the new `dbgsym`.

2.2.9 New method for naming network interfaces

The installer and newly installed systems will use a new standard naming scheme for network interfaces instead of `eth0`, `eth1`, etc. The old naming method suffered from enumeration race conditions that made it possible for interface names to change unexpectedly and is incompatible with mounting the root filesystem read-only. The new enumeration method relies on more sources of information, to produce a more repeatable outcome. It uses the firmware/BIOS provided index numbers and then tries PCI card slot numbers, producing names like `ens0` or `enp1s1` (ethernet) or `wlp3s0` (wlan). USB devices, which can be added to the system at any time, will have names based upon their ethernet MAC addresses.

This change does not apply to upgrades of jessie systems; the naming will continue to be enforced by `/etc/udev/rules.d/70-persistent-net.rules`. For more information, see `/usr/share/doc/udev/README.Debian.gz` or the [upstream documentation](https://www.freedesktop.org/wiki/Software/systemd/PredictableNetworkInterfaceNames/) (<https://www.freedesktop.org/wiki/Software/systemd/PredictableNetworkInterfaceNames/>).

2.2.10 News from Debian Med Blend

Besides several new packages and updates for software targeting life sciences and medicine, the Debian Med team has again put a focus on the quality of the provided packages. In a GSoC project and an Outreachy project, two students worked hard to add Continuous Integration support to the packages with the highest popularity-contest usage statistics. The latest Debian Med sprint in Bucharest also concentrated on package testing.

To install packages maintained by the Debian Med team, install the metapackages named `med-*`, which are at version 3.0.1 for Debian stretch. Feel free to visit the [Debian Med tasks pages](http://blends.debian.org/med/tasks) (<http://blends.debian.org/med/tasks>) to see the full range of biological and medical software available in Debian.

2.2.11 The Xorg server no longer requires root

In the stretch version of Xorg, it is possible to run the Xorg server as a regular user rather than as root. This reduces the risk of privilege escalation via bugs in the X server. However, it has some requirements for working:

- It needs **logind** and `libpam-systemd`.
- The system needs to support Kernel Mode Setting (KMS). Therefore, it may not work in some virtualization environments (e.g. `virtualbox`) or if the kernel has no driver that supports your graphics card.

- It needs to run on the virtual console it was started from.
- Only the `gdm3` display manager supports running X as a non-privileged user in stretch. Other display managers will always run X as root. Alternatively, you can also start X manually as a non-root user on a virtual terminal via **startx**.

When run as a regular user, the Xorg log will be available from `~/.local/share/xorg/`.

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 stretch can be found together with the Installation Guide on the [Debian website](https://www.debian.org/releases/stretch/debian-installer/) (<https://www.debian.org/releases/stretch/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](https://www.debian.org/releases/stretch/debian-installer/index#errata) (<https://www.debian.org/releases/stretch/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 previous official release with Debian 8, 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 jessie, please check the release announcements for the stretch beta and RC releases available from the Debian Installer's [news history](https://www.debian.org/devel/debian-installer/News/) (<https://www.debian.org/devel/debian-installer/News/>).

3.1.1 Major changes

Removed ports Support for the `powerpc` architecture has been removed.

New ports Support for the `mips64el` architecture has been added to the installer.

Desktop selection Since jessie, the desktop can be chosen within `tasksel` during installation, and several desktops can be selected at the same time.

New languages Thanks to the huge efforts of translators, Debian can now be installed in 75 languages, including English. 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.

The languages that can only be selected using the graphical installer as their character sets cannot be presented in a non-graphical environment are: Amharic, Bengali, Dzongkha, Gujarati, Hindi, Georgian, Kannada, Khmer, Malayalam, Marathi, Nepali, Punjabi, Tamil, Telugu, Tibetan, and Uyghur.

UEFI boot The stretch installer improves support for a lot of UEFI firmware and also supports installing on 32-bit UEFI firmware with a 64-bit kernel.

Note that this does not include support for UEFI Secure Boot.

New method for naming network interfaces The installer and the installed systems use a new standard naming scheme for network interfaces. `ens0` or `enp1s1` (ethernet) or `wlp3s0` (wlan) will replace the legacy `eth0`, `eth1`, etc. See Section 2.2.9 for more information.

Multi-arch images now default to amd64 Since 64-bit PCs have become more common, the default architecture on multi-arch images is now `amd64` instead of `i386`.

Full CD sets removed The full CD sets are not built anymore. The DVD images are still available as well as the `netinst` CD image.

Also, as the installer now gives an easy choice of desktop selection within `tasksel`, only `Xfce CD#1` remains as a single-CD desktop system.

Accessibility in the installer and the installed system The installer produces two beeps instead of one when booted with `grub`, so users can tell that they have to use the `grub` method of editing entries.

`MATE` desktop is the default desktop when `brltty` or `espeakup` is used in `debian-installer`.

Added HTTPS support Support for HTTPS has been added to the installer, enabling downloading of packages from HTTPS mirrors.

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 `jessie` installer, you cannot expect these to work with the new installer without modification.

The **Installation Guide** (<https://www.debian.org/releases/stretch/installmanual>) has an updated separate appendix with extensive documentation on using preconfiguration.

Chapter 4

Upgrades from Debian 8 (jessie)

4.1 Preparing for the upgrade

We suggest that before upgrading you also read the information in Chapter 5. That chapter covers potential issues which are not directly related to the upgrade process but 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 stretch, so a reboot will be necessary. Typically, this will be done after the upgrade is finished.

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 spends answering any configuration questions from package upgrades. Notice that if the upgrade process is left unattended and the system requests input during 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, and then upgrade the packages associated with your critical services. Upgrade these packages prior to doing the full upgrade described in Section 4.4.5. 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

Although Debian tries to ensure that your system stays bootable at all times, there is always a chance that you may experience problems rebooting your system after the upgrade. Known potential issues are documented in this and the next chapters of these Release Notes.

For this 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 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, 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.

For emergency recovery we generally recommend using the *rescue mode* of the stretch Debian Installer. The advantage of using the installer is that you can choose between its many methods to find one that best suits your situation. For more information, please consult the section “Recovering a Broken System” in chapter 8 of the [Installation Guide](https://www.debian.org/releases/stretch/installmanual) (<https://www.debian.org/releases/stretch/installmanual>) and the [Debian Installer FAQ](https://wiki.debian.org/DebianInstaller/FAQ) (<https://wiki.debian.org/DebianInstaller/FAQ>).

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.

4.1.4.1 Debug shell during boot using `initrd`

The `initramfs-tools` package includes a debug shell³ in the `initrds` 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.

4.1.4.2 Debug shell during boot using `systemd`

If the boot fails under `systemd`, it is possible to obtain a debug root shell by changing the kernel command line. If the basic boot succeeds, but some services fail to start, it may be useful to add `systemd.unit=rescue.target` to the kernel parameters.

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, especially 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.

Otherwise, the kernel parameter `systemd.unit=emergency.target` will provide you with a root shell at the earliest possible point. However, this is done before mounting the root file system with read-write permissions. You will have to do that manually with:

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

More information on debugging a broken boot under `systemd` can be found in the [Diagnosing Boot Problems](http://freedesktop.org/wiki/Software/systemd/Debugging/) (<http://freedesktop.org/wiki/Software/systemd/Debugging/>) article.

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.

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.

4.2 Checking system status

The upgrade process described in this chapter has been designed for upgrades from “pure” jessie 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.

Below there are two methods for finding such packages by using either `aptitude` or `apt-forktracer`. Please note that neither of them are 100% accurate (e.g. the `aptitude` example will list packages that were once provided by Debian but no longer are, such as old kernel packages).

```
$ aptitude search '~i(!~ODebian)'
$ apt-forktracer | sort
```

Direct upgrades from Debian releases older than 8 (jessie) are not supported. Please follow the instructions in the [Release Notes for Debian 8](https://www.debian.org/releases/jessie/releasenotes) (<https://www.debian.org/releases/jessie/releasenotes>) to upgrade to 8 first.

This procedure also assumes your system has been updated to the latest point release of jessie. 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 *jessie* and not to *stable* or *stretch*; see Section A.2.

To perform this review, launch **aptitude** in full-terminal 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` and `/etc/apt/preferences.d/`) 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 **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 *jessie* 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

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 stretch 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 jessie 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.

4.3 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 remote mirrors).

A release can often be referred to both by its codename (e.g. `jessie`, `stretch`) 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.

Debian provides two announcement mailing lists to help you stay up to date on relevant information related to Debian releases:

- By [subscribing to the Debian announcement mailing list](https://lists.debian.org/debian-announce/) (<https://lists.debian.org/debian-announce/>) you will receive a notification every time Debian makes a new release. Such as when `stretch` changes from e.g. `stable` to `oldstable`.
- By [subscribing to the Debian security announcement mailing list](https://lists.debian.org/debian-security-announce/) (<https://lists.debian.org/debian-security-announce/>), you will receive a notification every time Debian publishes a security announcement.

4.3.1 Adding APT Internet sources

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

Debian mirror addresses can be found at <https://www.debian.org/distrib/ftplist> (look at the “list of Debian mirrors” section). Note that FTP mirrors are being discontinued - see Section A.2.

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

```
http://mirrors.kernel.org/debian/dists/stretch/main/binary-mips/...
http://mirrors.kernel.org/debian/dists/stretch/contrib/binary-mips/...
```

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

```
deb http://mirrors.kernel.org/debian stretch 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 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.

4.3.2 Adding APT sources for a local mirror

Instead of using HTTP 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/local/debian/`, and have main directories like this:

```
/var/local/debian/dists/stretch/main/binary-mips/...
/var/local/debian/dists/stretch/contrib/binary-mips/...
```

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

```
deb file:/var/local/debian stretch 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 sources from optical media

If you want to use *only* CDs (or DVDs or Blu-ray Discs), 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 `/media/cdrom` mount point. For example, if `/dev/sr0` is your CD-ROM drive, `/etc/fstab` should contain a line like:

```
/dev/sr0 /media/cdrom auto noauto,ro 0 0
```

Note that there must be *no spaces* between the words `noauto,ro` in the fourth field.

To verify it works, insert a CD and try running

```
# mount /media/cdrom # this will mount the CD to the mount point
# ls -alF /media/cdrom # this should show the CD's root directory
# umount /media/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 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 proven 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 “stretch” or to “stable”. There should not be any sources entries pointing to jessie.

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-stretchstep.time -a ~/upgrade-stretchstep.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-stretch.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-stretch.time ~/upgrade-stretch.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 will have sufficient hard disk space when you start the full system upgrade described in Section 4.4.5. 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 is difficult to recover from.

apt-get can show you detailed information about 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 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 used **aptitude** or **apt-get** to manually install packages in jessie it will have kept track of those packages you manually installed, and will be able to mark as redundant those packages pulled in by dependencies alone which are no longer needed due to a package being removed. They will not mark for removal packages that you manually installed. To remove automatically installed packages that are no longer used, run:

```
# apt-get autoremove
```

You can also use **deborphan**, **debfooster**, or **cruft** to find redundant packages. Do not blindly remove the packages these tools present, especially if you are using aggressive non-default options that are prone to false positives. It is highly recommended that you manually review the packages suggested for removal (i.e. their contents, sizes, and descriptions) before you remove them.

- Remove packages that take up too much space and are not currently needed (you can always reinstall them after the upgrade). If you have `popularity-contest` installed, you can use **popcon-largest-unused** to list the packages you do not use that occupy the most space. You can find the packages that just take up the most disk space with **dpigs** (available in the `debian-goo` package) or with **wajig** (running `wajig size`). They can also be found with `aptitude`. Start **aptitude** in full-terminal mode, select Views → New Flat Package List, press `l` and enter `~i`, then press `S` and enter `~installsize`. This will give you a handy 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 `jessie` 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 4.4.5.

To do this, first run:

```
# apt-get upgrade
```

NOTE



The upgrade process for some previous releases recommended the use of **aptitude** for the upgrade. This tool is not recommended for upgrades from `jessie` to `stretch`.

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.

If the `apt-listchanges` package is installed, it will (in its default configuration) show important information about upgraded packages in a pager. Press `q` after reading to exit the pager and continue the upgrade.

4.4.5 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 some previous releases recommended the use of **aptitude** for the upgrade. This tool is not recommended for upgrades from `jessie` to `stretch`.

This will perform a complete upgrade of the system, installing the newest available versions of all packages, and resolving 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 CDs/DVDs/BDs, you will probably be asked to insert specific discs at several points during the upgrade. You might have to insert the same disc multiple times; this is due to inter-related packages that have been spread out over the discs.

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 install package`.

4.5 Possible issues during upgrade

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

4.5.1 Dist-upgrade fails with “Could not perform immediate configuration”

In some cases the **apt-get dist-upgrade** step can fail after downloading packages with:

```
E: Could not perform immediate configuration on 'package'. Please see man 5 apt. ←
  conf under APT::Immediate-Configure for details.
```

If that happens, running **apt-get dist-upgrade -o APT::Immediate-Configure=0** instead should allow the upgrade to proceed.

Another possible workaround for this problem is to temporarily add both jessie and stretch sources to your `sources.list` and run **apt-get update**.

4.5.2 Expected removals

The upgrade process to stretch might ask for the removal of packages on 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. For more information about packages obsoleted in stretch, see Section 4.8.

4.5.3 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
```

4.5.4 File conflicts

File conflicts should not occur if you upgrade from a “pure” jessie 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.

4.5.5 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.

4.5.6 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 may happen in desktop systems when the display manager is restarted.

To recover the console where the upgrade was running you will have to use `Ctrl + Alt + F1` (if in the graphical startup screen) or `Alt + F1` (if in the local text-mode console) to switch back to the virtual terminal 1. Replace `F1` with the function key with the same number as 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.

4.6 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 a kernel metapackage

When you dist-upgrade from `jessie` to `stretch`, it is strongly recommended that you install a `linux-image-*` metapackage, if you have not done so before. These metapackages will automatically pull in a newer version of the kernel during upgrades. You can verify whether you have one installed by running:

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

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

```
# apt-cache search linux-image- | grep -i meta | 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 "2.6.32-5-amd64", it is recommended that you install `linux-image-amd64`. 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-amd64
```

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. However, please have a look at Section 5.1.4 before performing the first reboot after the upgrade.

For the more adventurous there is an easy way to compile your own custom kernel on Debian. Install the kernel sources, provided in the `linux-source` package. You can make use of the `deb-pkg` target available in the sources' makefile for building a binary package. More information can be found in the [Debian Linux Kernel Handbook](http://kernel-handbook.alioth.debian.org/) (<http://kernel-handbook.alioth.debian.org/>), which can also be found as the `debian-kernel-handbook` package.

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.

4.7 Preparing for the next release

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

- Remove newly redundant or obsolete packages as described in Section 4.4.3 and Section 4.8. You should review which configuration files they use and consider purging the packages to remove their configuration files. See also Section 4.7.1.

4.7.1 Purging removed packages

It is generally advisable to purge removed packages. This is especially true if these have been removed in an earlier release upgrade (e.g. from the upgrade to `jessie`) or they were provided by third-party vendors. In particular, old `init.d` scripts have been known to cause issues.

CAUTION



Purging a package will generally also purge its log files, so you might want to back them up first.

The following command displays a list of all removed packages that may have configuration files left on the system (if any):

```
# dpkg -l | awk '/^rc/ { print $2 }'
```

The packages can be removed by using `apt-get purge`. Assuming you want to purge all of them in one go, you can use the following command:

```
# apt-get purge $(dpkg -l | awk '/^rc/ { print $2 }')
```

If you use `aptitude`, you can also use the following alternative to the commands above:

```
$ aptitude search '~c'
$ aptitude purge '~c'
```

4.8 Obsolete packages

Introducing lots of new packages, `stretch` also retires and omits quite a few old packages that were in `jessie`. 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 `stretch`'s release⁵, and will not normally provide other support in the meantime. Replacing them with available alternatives, if any, is recommended.

⁵ 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.

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); or they are no longer considered suitable for stretch 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.

The **Debian Bug Tracking System** (<https://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** (<https://bugs.debian.org/cgi-bin/pkgreport.cgi?pkg=ftp.debian.org&archive=yes>).

For a list of obsolete packages for Stretch, please refer to Section 5.1.3.

4.8.1 Dummy packages

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

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 Upgrade specific items for stretch

This section covers items related to the upgrade from jessie to stretch.

5.1.1 Late mounting of /usr is no longer supported

NOTE



This section only applies to systems using a custom kernel, where /usr is on a separate mount point from /. If you use the kernel packages provided by Debian, you are unaffected by this issue.

Mounting of /usr using only tools found in / is no longer supported. This has only worked for a few specific configurations in the past, and now they are explicitly unsupported.

This means that for stretch all systems where /usr is a separate partition need to use an initramfs generator that will mount /usr. All initramfs generators in stretch do so.

5.1.2 FTP access to Debian hosted mirrors will be removed

Debian hosted mirrors will stop providing FTP access. If you have been using the ftp: protocol in your sources.list, please migrate to http:. Please consider the following example for migrating:

```
deb http://deb.debian.org/debian stretch main
deb http://deb.debian.org/debian-security stretch/updates main

# tor variant (requires apt-transport-tor)
# deb tor+http://vwakviie2ienjx6t.onion/debian stretch main
# deb tor+http://sgvtcaew4bxjd7ln.onion/debian-security stretch/updates main
```

The above examples do not include non-free and contrib. Please remember to include these if you require those components enabled.

For more information, please refer to the announcement: [Shutting down public FTP services](https://www.debian.org/News/2017/20170425) (<https://www.debian.org/News/2017/20170425>).

5.1.3 Noteworthy obsolete packages

The following is a list of known and noteworthy obsolete packages (see Section 4.8 for a description).

The list of obsolete packages includes:

- Most `-dbg` packages have been removed from the main archive. They have been replaced by `-dbgsym` packages that are available from the `debian-debug` archive. Please see Section 2.2.8.
- The password managers `fpm2` and `kedpm` are no longer maintained upstream. Please use another password manager like `pass`, `keepassx`, or `keepass2`. Make sure that you extract your passwords from `fpm2` and `kedpm` before removing the packages.
- The `net-tools` package is being deprecated in favor of `iproute2`. See Section 5.3.9 or the [Debian reference manual](https://www.debian.org/doc/manuals/debian-reference/ch05#_the_low_level_network_configuration) (https://www.debian.org/doc/manuals/debian-reference/ch05#_the_low_level_network_configuration) for more information.
- The `nagios3` monitoring tools have been removed from stretch. The `icinga` package is the closest replacement. It reads its configuration files from a different path than `nagios` did, but is otherwise compatible.

5.1.4 Things to do post upgrade before rebooting

When `apt-get dist-upgrade` has finished, the “formal” upgrade is complete. For the upgrade to stretch, there are no special actions needed before performing a reboot.

5.1.5 Executables are now compiled as position independent executables (PIE) by default

By default, the GNU GCC 6 compiler provided by Debian stretch will compile all executables as position independent. This provides a mitigation for an entire class of vulnerabilities.

Unfortunately, the Linux kernel provided in Debian 8 (up to 8.7) has an issue that can cause some programs compiled as position independent executables to crash with a non-descriptive issue like `segmentation fault`. This issue is solved in the Linux version provided in 8.8 (version 3.16.43 or later) and in the kernel provided in Debian 9 (version 4.9 or later).

We recommend that you upgrade your kernel to a fixed version and then reboot before starting the upgrade to stretch. If you are running the kernel from Debian 8.8 or newer, you are not affected by this issue.

If you are *running* an affected version of the kernel during the upgrade, we highly recommend that you perform a reboot into the stretch kernel right after the upgrade to avoid hitting this.

5.1.5.1 Behavior changes of PIE for system administrators and developers

NOTE



This section is mainly intended for developers or system administrators. Desktop users are unlikely to be affected by this section.

The above also leads to some changes that are worth being aware of.

- The `file` tool (among others) will classify such binaries as “shared object” rather than an “executable”. If you have filters based on binary files, these may need to be updated (e.g. `spamfilters`).
- Static libraries being compiled into an executable now also need to be compiled as position independent code. The following error message from the linker is a symptom of this:

```
relocation ... against '[SYMBOL]' can not be used when making a shared
object; recompile with -fPIC
```

Note that even though the error message says `-fPIC`, it is sufficient to recompile with `-fPIE` (which is the default in the GCC 6 packages that are part of stretch).

- Historically, position independent executables have been associated with performance loss on some hardware. Notably the Debian architecture `i386` (32-bit Intel machines). While GCC 5 and GCC 6 have greatly **improved performance for position independent executables on 32-bit Intel** (<https://software.intel.com/en-us/blogs/2014/12/26/new-optimizations-for-x86-in-u>) this optimization may not be applicable to all architectures. Please consider evaluating the performance of your code if you are targeting machine architectures with a very limited number of registers.

5.1.6 Most LSB compatibility packages have been removed

Due to lack of interest and testability, Debian has removed the vast majority of the Linux Standard Base (LSB) compatibility packages.

Debian will still provide a selected few key LSB utilities used internally and externally, such as `lsb-release` and the `sysvinit` init functions in `lsb-base`. Furthermore, Debian is still firmly standing by the **Filesystem Hierarchy Standard (FHS) version 2.3 with the minor alterations described in the Debian Policy Manual** (<https://www.debian.org/doc/debian-policy/ch-opersys.html#s-fhs>).

5.1.7 32-bit MIPS now requires an R2 processor

The 32-bit MIPS support (both big- and little- endian) now requires a processor supporting MIPS32 Release 2 of the MIPS instruction set. Notably the Loongson-2E/2F and systems based on them (including the Yeeloong laptop) are no longer supported.

The following shell script can be used to indicate if your machine supports R2 (assuming only one type of processor is present). Note that Loongson-3 processors are supported even though they only claim to support MIPS32 Release 1.

```
if grep -E -q '^isa.*\bmips(32|64)r2\b' /proc/cpuinfo; then
    echo "OK (R2 supported)"
elif grep -q '^cpu model.*\bICT Loongson-3\b' /proc/cpuinfo; then
    echo "OK (Loongson 3)"
else
    echo "NOT OK: R2 not supported"
fi
```

5.2 Limitations in security support

There are some packages where Debian cannot promise to provide minimal backports for security issues. These are covered in the following subsections.

Note that the package `debian-security-support` helps to track the security support status of installed packages.

5.2.1 Security status of web browsers

Debian 9 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. Therefore, browsers built upon the `webkit`, `qtwebkit` and `khtml` engines are included in `stretch`, but not covered by security support. These browsers should not be used against untrusted websites.

For general web browser use we recommend Firefox or Chromium.

Chromium - while built upon the Webkit codebase - is a leaf package, which will be kept up-to-date by rebuilding the current Chromium releases for stable. Firefox and Thunderbird will also be kept up-to-date by rebuilding the current ESR releases for stable.

5.2.2 Lack of security support for the ecosystem around libv8 and Node.js

The Node.js platform is built on top of `libv8-3.14`, which experiences a high volume of security issues, but there are currently no volunteers within the project or the security team sufficiently interested and willing to spend the large amount of time required to stem those incoming issues.

Unfortunately, this means that `libv8-3.14`, `nodejs`, and the associated `node-*` package ecosystem should not currently be used with untrusted content, such as unsanitized data from the Internet.

In addition, these packages will not receive any security updates during the lifetime of the stretch release.

5.3 Package specific issues

In most cases, packages should upgrade smoothly between jessie and stretch. 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.

5.3.1 Older ciphers and SSH1 protocol disabled in OpenSSH by default

The OpenSSH 7 release has disabled some older ciphers and the SSH1 protocol by default. Please be careful when upgrading machines where you only have SSH access.

Moreover, the default of the "UseDNS" configuration option has changed from yes to no. This may cause users who use the "from=" functionality in `authorized_keys` to limit ssh access by host to be locked out, which is especially troublesome if upgrading remotely.

Please refer to the [OpenSSH documentation](http://www.openssh.com/legacy.html) (<http://www.openssh.com/legacy.html>) for more information.

5.3.2 Possible backwards incompatible changes to APT

This section covers some of the incompatible changes to APT that may affect your system.

5.3.2.1 APT now fetches files as an unprivileged user (`_apt`)

APT will now attempt to discard all root privileges before fetching files from mirrors. APT can detect some common cases where this will fail and fall back to fetching things as root with a warning. However, it may fail to detect some exotic setups (e.g. UID-specific firewall rules).

If you experience issues with this feature, please change to the `_apt` user and check that it:

- has read access to files in `/var/lib/apt/lists` and `/var/cache/apt/archives`.
- has read access to the APT trust store (`/etc/apt/trusted.gpg` and `/etc/apt/trusted.gpg.d/`)
- can resolve DNS names and download files. Example methods for testing:

```
# From the dnsutils package (if using tor, please check with tor-resolve ↵
instead).
$ nslookup debian.org >/dev/null || echo "Cannot resolve debian.org"
$ wget -q https://debian.org/ -O- >/dev/null || echo "Cannot download index ↵
page of debian.org"
```

For DNS issues, please check that `/etc/resolv.conf` is readable.

5.3.2.2 New APT pinning engine

APT 1.1 introduced a new pinning engine that now matches the description in the manual page.

The old engine assigned one pin priority per package; the new one assigns pin priorities per version. It then picks the version with the highest pin that is not a downgrade or that has a pin > 1000.

This changes the effect of some pins, especially negative ones. Previously, pinning a version to -1 effectively prevented the package from being installed (the package pin was -1); it now only prevents the version of this package from being installed.

5.3.2.3 New requirements for APT repository

NOTE



This section only applies if you have (or intend to use) third-party repositories enabled or if you maintain an APT repository.

To improve download stability and ensure security of the downloaded content, APT now requires the following from an APT repository:

- The InRelease file must be available.
- All metadata must include at least SHA256 checksums of all items. This includes the GPG signature of the InRelease file.
- Signatures on the InRelease file should be done with a key size of 2048 bits or larger.

If you rely on a third-party repository that cannot comply with the above, please urge them to upgrade their repository. More information about the InRelease file can be found on the [Debian Wiki](https://wiki.debian.org/RepositoryFormat#A.22Release.22_files) (https://wiki.debian.org/RepositoryFormat#A.22Release.22_files).

5.3.3 Desktops will migrate to libinput Xorg driver

NOTE



This section is only relevant if you have tweaked or need to change the default Xorg input configuration.

In *jessie*, the default input driver for Xorg is the `evdev` driver. In *stretch*, the default has changed to `libinput`. If you have Xorg configuration that relies on the `evdev` driver, you will either have to convert it to the `libinput` driver or reconfigure your system to use the `evdev` driver.

The following is an example configuration for `libinput` to enable the “Emulate3Buttons” feature.

```
Section "InputClass"
    Identifier "mouse"
    MatchIsPointer "on"
    Driver "libinput"
    Option "MiddleEmulation" "on"
EndSection
```

Insert it into `/etc/X11/xorg.conf.d/41-middle-emulation.conf`, reboot (or restart your Xserver) and it should now be enabled.

The `evdev` driver is still available in the `xserver-xorg-input-evdev` package.

5.3.4 Upstart removed

Due to the lack of upstream maintainers, the Upstart init system has been removed from *stretch*. If your system relies on this package, you should note that it will not be updated during the lifetime of Debian 9, and starting from Debian 10 (*buster*), Upstart jobs may be removed from packages.

Please consider switching to a supported init system, like `systemd` or `OpenRC`.

5.3.5 The debhelper tool now generates dbgsym packages by default

NOTE



This section is mainly intended for developers or organizations that build their own debian packages.

The debhelper tool suite will now generate `dbgsym` packages by default for ELF binaries. If you develop and package binaries, please check that your tooling supports these extra auto-generated packages.

If you use `reprepro`, you want to upgrade it to at least version 4.17.0. For `aptly`, you will need at least version 1.0.0, which is unfortunately not available in Debian stretch.

Should your tooling be unable to cope with these gracefully, you can ask debhelper to disable this feature by adding “`noautodbgsym`” in the `DEB_BUILD_OPTIONS` variable of your build service. Please see [the `dh_strip` manpage for more information](https://manpages.debian.org/stretch/debhelper/dh_strip.1.en.html) (https://manpages.debian.org/stretch/debhelper/dh_strip.1.en.html).

5.3.6 OpenSSL related changes

The `openssl` application expects option arguments before non-option arguments. For example, this does not work anymore:

```
openssl dsaparam 2048 -out file
```

while this still does:

```
openssl dsaparam -out file 2048
```

The `openssl enc` command changed the default digest (used to create the key from passphrase) from MD5 to SHA256. The digest can be specified with the `-md` option in case old files need to be decrypted with newer OpenSSL (or the other way around).

The 3DES and RC4 ciphers are no longer available for TLS/SSL communication. Servers linked against OpenSSL can't offer them and clients can't connect to servers which offer only those. This means that OpenSSL and Windows XP share no common cipher.

The package `libssl-dev` provides header files to compile against OpenSSL 1.1.0. The API changed a lot and it is possible that the software won't compile anymore. There is an [overview of the changes](https://wiki.openssl.org/index.php/1.1_API_Changes) (https://wiki.openssl.org/index.php/1.1_API_Changes). If you can't update your software, there is also `libssl1.0-dev` which provides headers against OpenSSL 1.0.2.

5.3.7 Perl changes that may break third-party software

NOTE



This section applies to code maintained outside Debian - local, third-party, or legacy Perl scripts and modules.

- Some modules have been removed from Perl core and are now shipped in separate packages. Notable examples are `CGI`, available in the `libcgi-pm-perl` package, and `Module::Build`, available in the `libmodule-build-perl` package.
- The current working directory (`.`) has been removed from the default list of include directories, `@INC`. This may affect usage of `require()`, `do()`, etc., where the arguments are files in the current directory.

All perl programs and module shipped by Debian should have been fixed to address any incompatibilities caused by the above; please file bugs if this is not the case. As the change has now been made in perl 5.26.0, third-party software should also start to be fixed. Information about how to fix this issue for developers is provided in the [perl 5.26 release notes](https://metacpan.org/changes/release/XSAWYERX/perl-5.26.0#Removal-of-the-current-directory-(%22.%22)-from-@INC) (https://metacpan.org/changes/release/XSAWYERX/perl-5.26.0#Removal-of-the-current-directory-(%22.%22)-from-@INC) (see the SECURITY section).

If needed you can temporarily reinstate `. in @INC` globally by commenting out the line in `/etc/perl/sitecustomize.pl` but you should only do this with a understanding of the potential risks. This workaround will be removed in Debian 10. You can also set the `PERL_USE_UNSAFE_INC` environment variable in a specific context which will have the same effect.

- The full list of changes in Perl since the version in Debian 8 is available in [perl522delta](https://metacpan.org/pod/release/RJBS/perl-5.22.0/pod/perldelta.pod) (https://metacpan.org/pod/release/RJBS/perl-5.22.0/pod/perldelta.pod) and [perl524delta](https://metacpan.org/pod/release/RJBS/perl-5.24.0/pod/perldelta.pod) (https://metacpan.org/pod/release/RJBS/perl-5.24.0/pod/perldelta.pod).

5.3.8 PostgreSQL PL/Perl incompatibility

The PostgreSQL PL/Perl procedural language package in jessie is incompatible with the Perl version in stretch. The `postgresql-plperl-9.4` package will be removed during the update, rendering server-side Perl procedures dysfunctional. Upgrading to PostgreSQL 9.6 should be unaffected; the procedures will work in the new PostgreSQL cluster if the `postgresql-plperl-9.6` package is installed. If unsure, take a backup of your PostgreSQL 9.4 clusters before upgrading to stretch.

5.3.9 net-tools will be deprecated in favor of iproute2

The `net-tools` package is no longer part of new installations by default, since its priority has been lowered from important to optional. Users are instead advised to use the modern `iproute2` toolset (which has been part of new installs for several releases already). If you still prefer to continue using the `net-tools` programs you can simply install it via

```
apt install net-tools
```

WARNING



Please keep in mind that `net-tools` may be uninstalled during the upgrade if it was only installed to satisfy a dependency. If you rely on `net-tools`, please remember to mark it as a manual installed package before the upgrade via:

```
apt-mark manual net-tools
```

Here is a summary of the `net-tools` commands, together with their `iproute2` equivalent:

legacy net-tools commands	iproute2 replacement commands
arp	ip n (ip neighbor)
ifconfig	ip a (ip addr), ip link, ip -s (ip -stats)
iptunnel	ip tunnel
nameif	ip link
netstat	ss, ip route (for netstat -r), ip -s link (for netstat -i), ip maddr (for netstat -g)
route	ip r (ip route)

5.3.10 The `_netdev` mount option is recommended when using AoE (ATA over ethernet) devices

NOTE



This only applies to systems that have ATA over ethernet (AoE) devices mounted. If the system does not mount any network shares, you can safely skip this section.

Due to a cleanup in the handling of network deconfiguration, AoE devices in use are no longer handled as expected during shutdown, possibly resulting in hangs and/or data loss. To mitigate that situation, it is suggested to mount such devices using the `_netdev` mount option. That option is available when using swap over AoE as well.

5.3.11 Harmless “Unescaped ... in regex is deprecated, ...” warnings during upgrade

During the upgrade, you may see some warning like:

```
Unescaped left brace in regex is deprecated, passed through in regex; marked by ←
<-- HERE in m/^(.*?) (\\)?\${ <-- HERE ([^{}]+)}(.*)$/ at /usr/share/perl5/ ←
Debconf/Question.pm line 72.
Unescaped left brace in regex is deprecated, passed through in regex; marked by ←
<-- HERE in m/\${ <-- HERE ([^}]+)}/ at /usr/share/perl5/Debconf/Config.pm ←
line 30.
```

These are harmless and happens if `perl-base` is upgraded before the `debconf` package.

5.3.12 SELinux policy store migration

NOTE



This section only applies to system that is using SELinux, which is not enabled by default.

In stretch, the SELinux policy store have moved from `/etc/selinux/<policy_name>` to `/var/lib/selinux/<policy_name>`. Furthermore, the format used inside the store has changed.

The policies provided by Debian (from e.g. the `selinux-policy-default` package) will be migrated automatically. However, system specific policies need to be migrated manually.

The `semanage-utils` package provides the script `/usr/lib/selinux/semanage_migrate_store` to do this transition.

5.3.13 iSCSI Enterprise Target no longer supported

The iSCSI Enterprise Target (IET), packaged in the `iscsitarget` package in previous releases, is no longer in Debian, as it will not work with recent kernel versions, and the project has seen no development activity in recent years.

Users of IET are encouraged to switch to the LIO stack, which is fully supported in Debian stretch. The package `targetcli-fb` provides the configuration utility for the LIO iSCSI target.

As the LIO stack was developed independently of the IET, the configuration has to be migrated manually.

Chapter 6

More information on Debian

6.1 Further reading

Beyond these release notes and the installation guide, further documentation on Debian is available from the Debian Documentation Project (DDP), whose goal is to create high-quality documentation for Debian users and developers, such as the Debian Reference, Debian New Maintainers Guide, the Debian FAQ, and many more. For full details of the existing resources see the [Debian Documentation website](https://www.debian.org/doc/) (<https://www.debian.org/doc/>) and the [Debian Wiki](https://wiki.debian.org/) (<https://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, though these should only be considered after researching the issue in available documentation. This section provides a short introduction to these sources 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 <https://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 support and aid for 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](https://wiki.debian.org/DebianIRC) (<https://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 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 can be browsed at <https://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 e-mail 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 e-mail. You can find out 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** (<https://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** (<https://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** (<https://lists.debian.org/>) is also extremely helpful. To maintain Debian's high-quality distribution, **submit bugs** (<https://bugs.debian.org/>) and help developers track them down and fix them. The tool `how-can-i-help` helps you to find suitable reported bugs to work on. If you have a way with words then you may want to contribute more actively by helping to write **documentation** (<https://www.debian.org/doc/cvs>) or **translate** (<https://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** (<https://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** (<https://www.debian.org/devel/#projects>) which include ports to particular architectures and **Debian Pure Blends** (<https://wiki.debian.org/DebianPureBlends>) for specific user groups, among many others.

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

BD

Blu-ray Disc

CD

Compact Disc

CD-ROM

Compact Disc Read Only Memory

DHCP

Dynamic Host Configuration Protocol

DLBD

Dual Layer Blu-ray Disc

DNS

Domain Name System

DVD

Digital Versatile Disc

GIMP

GNU Image Manipulation Program

GNU

GNU's Not Unix

GPG

GNU Privacy Guard

LDAP

Lightweight Directory Access Protocol

LSB

Linux Standard Base

LVM

Logical Volume Manager

MTA

Mail Transport Agent

NBD

Network Block Device

NFS

Network File System

NIC

Network Interface Card

NIS

Network Information Service

PHP

PHP: Hypertext Preprocessor

RAID

Redundant Array of Independent Disks

SATA

Serial Advanced Technology Attachment

SSL

Secure Sockets Layer

TLS

Transport Layer Security

UEFI

Unified Extensible Firmware Interface

USB

Universal Serial Bus

UUID

Universally Unique Identifier

WPA

Wi-Fi Protected Access

Appendix A

Managing your jessie system before the upgrade

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

A.1 Upgrading your jessie system

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

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

A.2 Checking your sources list

If any of the lines in your `/etc/apt/sources.list` refer to “stable”, it effectively points to stretch already. 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 by following the procedure below.

If you have also already installed packages from stretch, there probably is not much point in installing packages from jessie 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:`, `deb https:`, `deb tor+http:`, `deb tor+https:` or `deb ftp:`¹ for a reference to “stable”. If you find any, change `stable` to `jessie`.

NOTE



Lines in `sources.list` starting with “`deb ftp:`” and pointing to `debian.org` addresses should be changed into “`deb http:`” lines. See Section 5.1.2.

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

¹ Debian will remove FTP access to all of its official mirrors on 2017-11-01 (<https://lists.debian.org/debian-announce/2017/msg00001.html>). If your `sources.list` contains a `debian.org` host, please consider switching to deb.debian.org (<https://deb.debian.org>). This note only applies to mirrors hosted by Debian itself. If you use a secondary mirror or a third-party repository, then they may still support FTP access after that date. Please consult with the operators of these if you are in doubt.

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 Removing obsolete configuration files

Before upgrading your system to stretch, it is recommended to remove old configuration files (such as `*.dpkg-{new,old}` files under `/etc`) from the system.

A.4 Upgrade legacy locales to UTF-8

Using a legacy non-UTF-8 locale has been unsupported by desktops and other mainstream software projects for a long time. Such locales should be upgraded by running **dpkg-reconfigure locales** and selecting a UTF-8 default. You should also ensure that users are not overriding the default to use a legacy locale in their 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, David Prévot, Eddy Petrișor, Emmanuel Kasper, Esko Arajärvi, Frans Pop, Giovanni Rapagnani, Gordon Farquharson, Javier Fernández-Sanguino Peña, Jens Seidel, Jonas Meurer, Jonathan Nieder, Joost van Baal-Ilić, Josip Rodin, Julien Cristau, Justin B Rye, LaMont Jones, Luk Claes, Martin Michlmayr, Michael Biebl, Moritz Mühlenhoff, Niels Thykier, Noah Meyerhans, Noritada Kobayashi, Osamu Aoki, Peter Green, Rob Bradford, Samuel Thibault, Simon Bienlein, Simon Pailard, 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

B

BIND, 4
Blu-ray, 4

C

Calligra, 4

D

DocBook XML, 2
dual layer Blu-ray, 4
DVD, 4

E

Evolution, 4
Exim, 4

G

GCC, 4
GNOME, 3
GnuPG, 4

I

Inkscape, 4

K

KDE, 3

L

LibreOffice, 4
LXDE, 3
LXQt, 3

M

MariaDB, 4
MATE, 3

N

Nginx, 4

O

OpenJDK, 4
OpenSSH, 4

P

packages
 apt, 1, 2, 5, 15, 16
 apt-listchanges, 19
 aptitude, 5, 18, 22
 aptly, 30
 dblatex, 2
 debconf, 32
 debian-goodies, 18
 debian-kernel-handbook, 22
 debian-security-support, 27
 default-mysql-*, 5
 default-mysql-client, 5
 default-mysql-server, 4
 doc-debian, 34

docbook-xsl, 2
dpkg, 1
fpm2, 26
gdm3, 7
gnupg, 5
gnupg1, 5
how-can-i-help, 34
icinga, 26
initramfs-tools, 12, 21
iproute2, 26, 31
iscsitarget, 32
kedpm, 26
keepass2, 26
keepassx, 26
libcgi-pm-perl, 30
libmodule-build-perl, 30
libpam-systemd, 6
libssl-dev, 30
libssl1.0-dev, 30
libv8-3.14, 28
linux-image-*, 21
linux-image-amd64, 21
linux-source, 22
localepurge, 18
lsb-base, 27
lsb-release, 27
mariadb-client-10.1, 5
mariadb-server-10.1, 4
mysql-defaults, 4
mysql-server-5.5, 4
mysql-server-5.6, 4
nagios3, 26
net-tools, 26, 31
nodejs, 28
pass, 26
perl-base, 32
pkg-dbg, 6
pkg-dbgsym, 6
popularity-contest, 18
postgresql-plperl-9.4, 31
postgresql-plperl-9.6, 31
release-notes, 1
reprepro, 30
selinux-policy-default, 32
semanage-utils, 32
targetcli-fb, 32
tinc, 13
udev, 21
upgrade-reports, 1
virtual-mysql-*, 5
xmlroff, 2
xserver-xorg-input-evdev, 29
xsltproc, 2
Perl, 4
PHP, 4
Postfix, 4

PostgreSQL, 4

X

Xfce, 3